CHANGE IMPACT ANALYSIS TO MANAGE PROCESS EVOLUTION IN WEB WORKFLOWS

Jeewani Anupama Ginige

A thesis submitted for the degree of Doctor of Philosophy

School of Computing and Mathematics
College of Health and Science
University of Western Sydney
Australia

2008

© Jeewani Anupama Ginige – 2008
I dedicate this achievement to

two most important people in my life, my husband
Viru and my little son Methin
(as the suffering of writing is for the others who patiently wait for you to finish)

&

my brother Ranga
(who wished for this achievement of mine, but never got to see it)
Acknowledgement

I would like to take this opportunity to reflect back on the journey in the last four years; during which I tried to balance the responsibilities and duties of a wife, a mother, an academic, and a research student. This journey would have not been possible without a number of people who have assisted me in countless ways.

Firstly, I would like to extend my sincere gratitude to Prof. Athula Ginige, who has been an excellent mentor for me. Prof. Ginige’s vision and patience in guiding me towards the desired goal is the foundation of this achievement. Specially thank you for those intellectual discussions that helped to shape my research and for the encouragement and support given to me in publishing my work.

My heartfelt appreciations to Dr. Robyn Lawson who agreed to join the supervisory panel at a late stage as the principle supervisor. Dr. Lawson’s dedication to guide and motivate her students, especially in writing stages of a thesis, is extraordinary. I also would like to thank both Dr. Lawson and Prof. Ginige for taking care of all the complexities of administrative tasks that are beyond my control and comprehension.

Special gratitude to Dr. Uma Srinivasan, who is also a member of the supervisory panel, for her excellent advice, which has helped me to formulate my research. I also would like to thank her for the suggestions on improving the readability of my work and guiding me in various ways to achieve this goal.

My gratitude to the Cisco Systems Australia Pty Ltd for the scholarship funding that released me from teaching duties in the early years of this research.

Many thanks to the initial steering committee of OCAS (Online Course Approval System) project at UWS, which included Prof. Anne Cusick, Prof. Geoff Scott, and Prof. Nigel Bond, for allowing me to use OCAS as a case study in my research. In addition, I also would like to extend my gratitude to former and present staff of the Academic Registrars unit who were involved in OCAS, including Kate Massey-Greene, Shirley Anderson, and Jeff Warnock. Also, sincere thanks to OCAS development team - Makis Marmaridis, Danny Liang, Vidura Dissanayake, and Buddhima De Silva, of the AeIMS research group for making the OCAS project a success and helping me in various ways to use OCAS as a case study in my work.
I also would like to thank other fellow researchers at SCM-Parramatta Ana, Joanne, Dinesh, Shiromi, Abbass, and Mahesha for sharing ideas in managing activities related to research. In addition, sincere appreciation to former and present administrative and technical staff at the SCM including Veena, Leif, Angeline, Guang, Rocky, Marianne, Claudia, Tracey, and Olga who have helped me in countless ways over the last four years. Special thanks to my former and present academic supervisors Paul, Graeme, Yan and Leanne for considering research activities when allocating teaching work. Further my great appreciation to fellow academic staff for helping me in various ways to manage teaching activities alongside research.

Special thanks to Nuwan Ginige who did an excellent job in proofreading the thesis.

My sincere thanks to all close family and friends that are too numerous to mention, who helped me in many ways. Your kindness in supporting me emotionally and in countless other ways helped me to manage my busy life. In addition, most deservingly many thanks to my mother and my late father for giving me a solid foundation in education. Further, thanks to all the teachers that have taught me in my life, as every piece of knowledge that I got from you has helped me to achieve this goal.

Finally but most importantly, my great appreciation to my loving husband Viru and my little son Methin. Viru has always encouraged me to achieve the best and helped me in various ways to accomplish this goal. Also, I would like to mention my little son Methin, who tried so hard to understand my absence and cope with it. With immense gratitude I mention, it is the love of you two that helped me to achieve this goal, amongst numerous hurdles. So thank you again Viru and Methin.
Statement of Authentication

The work presented in this thesis is, to the best of my knowledge and belief, original except as acknowledged in the text. I hereby declare that I have not submitted this material either in full or in part, for a degree at this or any other institution.

...........................................................................................................

Jeewani Anupama Ginige

...........................................................................................................

Date
Table of Contents

CHANGE IMPACT ANALYSIS TO MANAGE PROCESS EVOLUTION IN WEB WORKFLOWS

List of Tables .................................................................................................................. v
List of Figures ................................................................................................................ vi
List of KAT (Kleene Algebra with Test) Algebraic Expressions ..................................... ix
List of Abbreviations .................................................................................................... x
Abstract .......................................................................................................................... xii
List of Published Work .................................................................................................. xiv

1 INTRODUCTION .......................................................................................................... 1
1.1 MOTIVATION .............................................................................................................. 2
1.2 PARADIGM OF PROCESS AUTOMATION – PoPA FRAMEWORK ......................... 5
  1.2.1 Pragmatic Level - Contextual Information ...................................................... 6
  1.2.2 Semantic Level - Operational Business Processes .......................................... 6
  1.2.3 Syntactic Level - Business Process Models ..................................................... 8
  1.2.4 Implementation Level – Web-based Workflow Systems ................................... 9
  1.2.5 Role of the PoPA Framework in Process Evolution ...................................... 10
1.3 RESEARCH PROBLEM ............................................................................................. 11
1.4 CONTEXT OF THIS RESEARCH IN THE BROADER THEORETICAL FRAMEWORK ...... 13
1.5 CONTRIBUTIONS AND SIGNIFICANCE OF THIS RESEARCH .............................. 16
  1.5.1 Major Contributions ...................................................................................... 16
  1.5.2 Significance of this work in BPEM ................................................................. 17
1.6 THESIS STRUCTURE................................................................................................. 19
1.7 CHAPTER SUMMARY ............................................................................................... 21

2 RESEARCH METHODOLOGY .................................................................................... 23
  2.1 CHAPTER OVERVIEW ............................................................................................ 23
  2.2 THE RESEARCH FRAMEWORK ............................................................................ 24
  2.3 SELECTION PROCESS OF THE RESEARCH METHODOLOGY ....................... 27
    2.3.1 Research Philosophy .................................................................................... 27
    2.3.2 Research Approach ..................................................................................... 31
    2.3.3 Research Strategy ....................................................................................... 33
    2.3.4 Methods of Data Collection ....................................................................... 38
    2.3.5 Level of Applicability of the Selected Methodology ................................... 39
  2.4 APPLICATION OF THE SELECTED METHODOLOGY IN THIS RESEARCH ......... 40
  2.5 CHAPTER SUMMARY ............................................................................................ 40

3 LITERATURE REVIEW ............................................................................................... 42
  3.1 CHAPTER OVERVIEW ............................................................................................ 42
  3.2 KEY TERMINOLOGIES .......................................................................................... 44
    3.2.1 Process, Business Process, and Process Elements ....................................... 44
    3.2.2 Workflow, WfMS, Process Definitions, and Instances ............................... 48
    3.2.3 Pragmatic, Semantic, Syntactic, and Implementation of Processes .............. 50
      3.2.3.1 Contextual Artefacts in the Pragmatic Level ........................................ 50
      3.2.3.2 Operational Business Process in the Semantic Level ............................ 53
      3.2.3.3 Process Models in the Syntactic Level .................................................. 55
      3.2.3.4 Web Workflow Artefacts in the Implementation Level ........................... 56
  3.3 THEORETICAL FRAMEWORK .............................................................................. 56
    3.3.1 BPA and BPEM ............................................................................................. 59
      3.3.1.1 BPA Step – Process Identification ......................................................... 59
      3.3.1.2 BPA Step – Process Re-Design (Re-Engineering) ................................. 59
      3.3.1.3 BPA Step – Create Process Models ..................................................... 60
      3.3.1.4 BPA Step – Analyse Process Models (Workflow Analysis) .................. 60
      3.3.1.5 BPA Step – Implement Process Models .............................................. 61
      3.3.1.6 BPEM Step – Evolution Diagnosis .................................................... 61
      3.3.1.7 BPEM Step – Impact Analysis of Business Process Evolution .......... 61
# 6 KAT FOR CONCEPTUALISING CAD AMONG PROCESS ELEMENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Chapter Overview</td>
<td>197</td>
</tr>
<tr>
<td>6.2 Selection of a Suitable Formalism</td>
<td>200</td>
</tr>
<tr>
<td>6.3 Kleene Algebra with Tests (KAT)</td>
<td>203</td>
</tr>
<tr>
<td>6.4 KAT for Modelling Business Processes</td>
<td>206</td>
</tr>
<tr>
<td>6.4.1 Representing Associations among Actions using KAT</td>
<td>206</td>
</tr>
<tr>
<td>6.4.1.1 Application of KAT Definitions-1 to 8 in OCAS</td>
<td>209</td>
</tr>
<tr>
<td>6.4.2 Representing Associations between Actions and Participants using KAT</td>
<td>213</td>
</tr>
<tr>
<td>6.4.2.1 Application of KAT Definitions-9, 10 and 12 in OCAS</td>
<td>214</td>
</tr>
<tr>
<td>6.4.3 Representing Associations between Actions and Business Object using KAT</td>
<td>217</td>
</tr>
<tr>
<td>6.4.3.1 Application of KAT Definitions-12, 13 and 14 in OCAS</td>
<td>219</td>
</tr>
<tr>
<td>6.4.4 Representing Associations among Business Object Elements using KAT</td>
<td>226</td>
</tr>
<tr>
<td>6.4.4.1 Application of KAT Definitions-15 and 16 in OCAS</td>
<td>227</td>
</tr>
<tr>
<td>6.4.5 Representing Constraints on the Associations among Actions using KAT</td>
<td>231</td>
</tr>
<tr>
<td>6.4.5.1 Application of KAT Definitions-17 and 18 in OCAS</td>
<td>232</td>
</tr>
<tr>
<td>6.4.6 Representing Constraints on the Associations between Actions and Participants using KAT</td>
<td>234</td>
</tr>
<tr>
<td>6.4.6.1 Application of KAT Definitions-19 and 20 in OCAS</td>
<td>236</td>
</tr>
<tr>
<td>6.4.7 Representing Constraints on the Associations between Actions and Business Object Elements using KAT</td>
<td>238</td>
</tr>
<tr>
<td>6.4.8 Representing Constraints on the Associations among Business Object Elements using KAT</td>
<td>238</td>
</tr>
<tr>
<td>6.5 KAT Expression of the New Course Approval Process</td>
<td>239</td>
</tr>
<tr>
<td>6.6 Data Structure to Capture KAT Expressions</td>
<td>244</td>
</tr>
<tr>
<td>6.7 Chapter Summary</td>
<td>248</td>
</tr>
</tbody>
</table>

# 7 Nature of Process Element Evolution

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1 Chapter Overview</td>
<td>249</td>
</tr>
<tr>
<td>7.2 Characterisation of Process Elements</td>
<td>251</td>
</tr>
<tr>
<td>7.3 Semantics of Process Element Evolution</td>
<td>253</td>
</tr>
<tr>
<td>7.3.1 Action Evolution Semantics</td>
<td>254</td>
</tr>
<tr>
<td>7.3.2 Participant Evolution Semantics</td>
<td>258</td>
</tr>
<tr>
<td>7.3.2.1 Additions and Deletions of Participants</td>
<td>258</td>
</tr>
<tr>
<td>7.3.2.2 Modification of Participants</td>
<td>259</td>
</tr>
<tr>
<td>7.3.3 Business Data Object Evolution Semantics</td>
<td>260</td>
</tr>
<tr>
<td>7.3.3.1 Addition and Deletion of Business Data Object Elements</td>
<td>261</td>
</tr>
<tr>
<td>7.3.3.2 Modification of Business Data Object Elements</td>
<td>261</td>
</tr>
<tr>
<td>7.4 Dynamics of Process Element Evolution</td>
<td>263</td>
</tr>
<tr>
<td>7.4.1 Example of Dynamics of Process Evolution</td>
<td>265</td>
</tr>
<tr>
<td>7.5 DISN Impacts of Process Element Changes</td>
<td>271</td>
</tr>
<tr>
<td>7.6 Chapter Summary</td>
<td>276</td>
</tr>
</tbody>
</table>

# 8 Process Evolution and Change Impact Analysis (PECIA) Model

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1 Chapter Overview</td>
<td>278</td>
</tr>
<tr>
<td>8.2 Evolution Initiation and Propagation</td>
<td>279</td>
</tr>
<tr>
<td>8.3 Process Evolution and Change Impact Analysis (PECIA) Model</td>
<td>281</td>
</tr>
<tr>
<td>8.3.1 Methodology for using the PECIA Model</td>
<td>283</td>
</tr>
<tr>
<td>8.4 Using the PECIA Model to Locate Propagating Impact</td>
<td>284</td>
</tr>
<tr>
<td>8.4.1 Handling Changes at the Pragmatical level</td>
<td>285</td>
</tr>
<tr>
<td>8.4.1.1 Step 1 - Changes to Contextual Artefacts</td>
<td>285</td>
</tr>
<tr>
<td>8.4.1.2 Step 2 - Identify Changing Contextual Artefacts</td>
<td>286</td>
</tr>
<tr>
<td>8.4.1.3 Step 3 - Temporary Record of Contextual Artefact Changes</td>
<td>286</td>
</tr>
<tr>
<td>8.4.1.4 Step 4 - Find propagating impact on other Contextual Artefacts</td>
<td>286</td>
</tr>
<tr>
<td>8.4.1.5 Step 5 - Prompting other Impacted Contextual Artefacts</td>
<td>286</td>
</tr>
<tr>
<td>8.4.2 Handling Changes at the Semantic level</td>
<td>287</td>
</tr>
<tr>
<td>8.4.2.1 Step 6 - Finding Process Elements affected by Context Changes</td>
<td>288</td>
</tr>
<tr>
<td>8.4.2.2 Step 7 - Changes to Operational Processes</td>
<td>290</td>
</tr>
<tr>
<td>8.4.2.3 Step 8 - Identify Changing Process Elements</td>
<td>290</td>
</tr>
<tr>
<td>8.4.2.4 Step 9 - Temporary Record of Process Element Change</td>
<td>290</td>
</tr>
<tr>
<td>8.4.2.5 Step 10 - Finding the Propagating Impact of the Element Change</td>
<td>291</td>
</tr>
<tr>
<td>8.4.2.6 Step 11 - List of Impacted Process Elements</td>
<td>291</td>
</tr>
<tr>
<td>8.4.3 Handling Changes at the Syntactic Level</td>
<td>291</td>
</tr>
</tbody>
</table>
9 DISCUSSION, CONCLUSION AND FUTURE RESEARCH DIRECTIONS ......................... 312

9.1 CHAPTER OVERVIEW ......................................................................................... 312
9.2 RESEARCH SUMMARY ..................................................................................... 313
9.3 CONTRIBUTIONS OF THIS RESEARCH .......................................................... 315
  9.3.1 Primary Contributions ................................................................................. 315
  9.3.2 Secondary Contributions .......................................................................... 317
9.4 PRACTICAL RELEVANCE ................................................................................ 318
9.5 THEORETICAL RELEVANCE .......................................................................... 319
  9.5.1 PECLA Model viewed as a Knowledge Repository .................................... 320
  9.5.2 Comparison with Similar Work ................................................................. 321
  9.5.2.1 Solution to a Process Modelling Challenge using KAT ....................... 322
9.6 LIMITATIONS .................................................................................................. 323
9.7 FUTURE RESEARCH DIRECTIONS ................................................................ 324
9.8 CHAPTER SUMMARY AND CONCLUSION ...................................................... 325
  REFERENCES .................................................................................................... 327
  BIBLIOGRAPHY ............................................................................................... 355
APPENDIX A ...................................................................................................... 360
APPENDIX B ...................................................................................................... 363
List of Tables

Table 2.1- Meta-Theoretical Assumptions in Positivism, Interpretivism and Critical Research - Adapted based on Weber (2004) and Orlikowski et al. (1991) ........................................ 29
Table 3.1- Summary of Literature in the area of Workflow Analysis and Propagating Impact Analysis .............................................................. 96
Table 3.2- Strengths and Weaknesses of Process Conceptualisation Models in Capturing Associations of Process Elements ...................................................... 104
Table 3.3- Capability of Formalisms to Represent Correlations of Process Elements .......... 111
Table 4.1- Contextual Information that defines the New Course Approval Process ............ 123
Table 4.2- Leakage of Information of the New Course Approval Process ..................... 127
Table 4.3- Full set of Actions and Participants of the New Course Approval Process ........ 133
Table 4.4- Evolutions and Changes that affected the OCAS Processes ....................... 149
Table 5.1- The ability of the Schema in Figure 5.2 to Capture Correlations among Contextual Information Elements ....................................................... 161
Table 5.2- Visibility Aspect of Business Object Attributes in an Action UI ..................... 171
Table 6.1- Comparison of the ability of three different Formalisms in capturing CAD among Process Elements .............................................................. 202
Table 6.2- Alphas related to the OCAS New Course Approval Process ......................... 209
Table 6.3- Association between Process Participants and Actions in the New Course Approval Process ................................................................. 214
Table 6.4- Description of Data visibility in action α1 - Create Course Notification of Intention (CNI) of the New Course Approval Process ......................................... 220
Table 6.5- Guard Conditions of the Action in which Course Information is initially entered in the New Course Approval Process ........................................... 222
Table 6.6- Data Interaction Dependency in the New Course Approval Process ............... 224
Table 6.7- Routing Conditions associated with the New Course Approval Process ........ 226
Table 6.8- The Integrity Constraints Associated with the Information Object of the New Course Approval Process ......................................................... 229
Table 6.9- Computational Constraints among information attributes in course object of the New Course Approval Process .................................................... 230
Table 6.10- Participant Comparison Constraints in the New Course Approval Process ..... 237
Table 6.11- The expanded Composite Conditions in the New Course Approval Process ... 239
Table 7.1- Attributes that Characterises Process Elements ........................................... 252
Table 7.2- Semantics of Process Action Evolution ...................................................... 255
Table 7.3- Dynamics of Process Evolution ............................................................... 267
Table 8.1- Decision Table for the Algorithmic-Step-3 ................................................ 305
Table 8.2- Using the PECIA Model to handle Evolution Scenario 1 of the New Course Approval Process of OCAS ................................................... 307
Table 8.3- Using the PECIA Model to handle Evolution Scenario 2 of the New Course Approval Process of OCAS ................................................... 309
Table 9.1- Validation of Practical Utility of Contributions in Relevant Chapters ............ 318
Table 9.2- Comparison of the Contributions of this Research with Similar Work ............ 321
List of Figures

Figure 1.1- The PoPA framework adapted from zur Muehlen (2001) and Georgakopoulos, Hornick, and Sheth (1995) ................................................................. 5

Figure 1.2- Theoretical framework in relation to BPA and BPEM - Adapted based on Kettinger et al. (1997) and Georgakopoulos et al. (1995) ......................... 14

Figure 1.3- Outline of the Chapters in this Thesis .......................................................... 20

Figure 2.1- Layers in Saunders et al.’s (2003) Research Process Onion .......................... 25

Figure 2.2- Combined use of both Inductive and Deductive Approaches in this Research adapted from Blaikie (1993) ................................................................. 33

Figure 2.3- Application of Constructive Research Steps in this Research. Extended from the work of Kasanen et al. (1993) ................................................................. 36

Figure 2.4- Summary of the Research Methodology Applicable to this Research .............. 39

Figure 2.5- Application of Selected Research Strategies in this Thesis ............................ 40

Figure 3.1- Workflow Positioned in the Hype-Curve, adopted from Baeyens (2004) ........ 42

Figure 3.2- The Dimensions of the Workflow Context Space by Maus (2001) ................... 51

Figure 3.3- Positioning of this Research in BPA and BPEM. Adapted from Kettinger et al. (1997) and Georgakopoulos et al. (1995) ............................................. 57

Figure 3.4- The Distributed METEOR2 Model of WebWorks (Miller et al., 1998) .......... 68

Figure 3.5- WWWorkflow System Architecture - adopted from Ames et al. (1997) .......... 68

Figure 3.6- ADOME-WFMS architecture - adopted from Chiu et al. (2001c) .................. 69

Figure 3.7- Logical Architecture of LBPF engine - adopted from Liang et al. (2007) ....... 70

Figure 3.8- Dependencies and Associations between Aspects in Web Application - adopted from De Silva et al. (2007) ................................................................. 71

Figure 3.9- Workflow Reference Model (WfMC, 1999) .................................................. 72

Figure 3.10- The Workflow Activity Model WAMO (Eder et al., 1995) ................. 75

Figure 3.11- Dynamic Workflow Management (DWM) Meta-Model (Kwan et al., 1997) .... 75

Figure 3.12- Three-Level Meta-Modelling for ADOME-WFMS (Chiu, Li, & Karlapalem, 1999) ................................................................. 76

Figure 3.13- WfMC’s Workflow Meta-Model (WfMC, 1999) ........................................ 77

Figure 3.14- Event-driven Process Chain (EPC) Meta-Model with Extension for Performance Evaluation (Stefanov et al., 2005) ........................................ 77

Figure 3.15- Meta-model for Case Handling Paradigm (van der Aalst et al., 2005d; van der Aalst, 2001b) ................................................................. 78

Figure 3.16- Example of WF-net (Petri-Nets based) Process Modelling Tool (van der Aalst, 2005) ................................................................. 78

Figure 3.17- An Example of Process Modelling using EPCs (Nüttgens et al., 1998) ....... 81

Figure 3.18- UML Activity Diagram with Activity Partitioning to Swim Lanes (OMG, 2007) ................................................................. 82

Figure 3.19- WF-Graphs for Process Modelling (Sadiq et al., 2000c) ....................... 83

Figure 3.20- BPMN Process Example (OMG, 2004) .................................................. 83

Figure 3.21- Dimensions of Static Evolution (van der Aalst et al., 1999) ....................... 91

Figure 4.1- High-level New Course Approval Process of OCAS- adopted from AUQA Report (AUQA, 2007) ................................................................. 126
Figure 5.25- Schema to capture CAD between Elements in the Syntactic and Implementation Levels.......................................................... 195
Figure 6.1- Sequential Composition between two Actions.................................................. 206
Figure 6.2- Conditional Choice between two Actions.......................................................... 207
Figure 6.3- Looping or Iteration of an Action until a Guard Condition is satisfied.............. 207
Figure 6.4 - Parallelism between two Actions................................................................. 208
Figure 6.5 - Actions grouped in to Sub-Processes in the New Course Approval process of OCAS...................................................................... 210
Figure 6.6- Sub Process Triggers that affect the Association between Actions in the New Course Approval Process of OCAS.................................................. 233
Figure 6.7- Binary Tree representation of main flow control patterns in KAT...................... 245
Figure 6.8- Binary Tree Structure to represent the KAT Expression 6.11 ......................... 247
Figure 7.1 - Compulsory Propagating Impact of Deletion of a Process Participant.............. 263
Figure 7.2 - Non-Compulsory Propagating Impact of Addition of a Process Participant..... 264
Figure 7.3- Dynamics of Addition, Deletion and Modification of Process Actions............. 264
Figure 7.4- Dynamics of Addition, Deletion and Modification of Process Participants..... 265
Figure 7.5- Dynamics of Addition, Deletion and Modification of Business Data Object Elements........................................................................ 265
Figure 7.6- Course Notification of Intention (CNI) and Business Case (BC) approval of new courses approval process in OCAS................................. 266
Figure 7.7- Hypothetical Leave Approval Process.............................................................. 272
Figure 7.8- DISN impact of Evolution Scenario 1.............................................................. 274
Figure 7.9- DISN impact of the Evolution Scenario 2...................................................... 275
Figure 8.1- Initiation of changes and Propagating Impact from Higher levels down to the implementation Level Artefacts........................................ 280
Figure 8.2- Process Evolution and Change Impact Analysis (PECIA) Model........................ 282
Figure 8.3- High-Level Methodology for using the PECIA Model for Business Process Management Purposes adapted based on Castano et al. (1999)........ 284
Figure 8.4- Steps in handling Changes at the Pragmatic level.......................................... 285
Figure 8.5- List of Associated contextual Artefacts............................................................ 287
Figure 8.6- Steps in handling Changes at the Semantic level.......................................... 288
Figure 8.7- List of process elements affected due to Contextual Artefact changes........... 289
Figure 8.8- Steps in handling Changes at the Syntactic level........................................... 292
Figure 8.9- Showing the models affected by the process element change...................... 293
Figure 8.10- List of other Model/s affected due a particular model change .................... 295
Figure 8.11- Steps in handling changes at the Implementation Level.............................. 296
Figure 8.12- Showing the Web Artefacts that are affected by a model change.................. 297
Figure 8.13- List of Web workflow Artefacts Possible to be affected............................ 299
Figure 8.14- Binary Tree Representation of KAT expression of a particular Process........... 303
Figure 9.1- Applicability of the Contributions in BPA and BPEM..................................... 320
Figure 9.2- Process Modelling Challenge set by van der Aalst (2003)............................. 323
List of KAT (Kleene Algebra with Test) Algebraic Expressions

**KAT Expression 6.1** - Sub Process of CNI creation in New course approval process ....... 211
**KAT Expression 6.2** - Process of FCP creation in New course approval process ............. 211
**KAT Expression 6.3** - Sub Process of course code entry in New course approval process 211
**KAT Expression 6.4** - Sub Process of internal forum report in New course approval process
................................................................. 211
**KAT Expression 6.5** - Sub Process of external advisory committee report in New course
approval process .................................................. 211
**KAT Expression 6.6** - Sub Process of Resource Assessment in New course approval process
............................................................................ 211
**KAT Expression 6.7** - Sub Process of Business Case Approval in New course approval
process ........................................................................ 211
**KAT Expression 6.8** - Composite statement of the New Course Approval Process .......... 213
**KAT Expression 6.9** - Expanded New course approval process, using all action notations 213
**KAT Expression 6.10** - The New Course Approval Process of OCAS presented as a KAT
Expression using alphas and phis ..................................... 243
**KAT Expression 6.11** - Sample KAT expression of a hypothetical process ...................... 247
**KAT Expression 7.1** - KAT Expression for the process in Figure 7.7 ......................... 274
List of Abbreviations

AeIMS - Advanced enterprise Information Management System
AS - Academic Senate
ASP - Active Server Pages
AUQA - Australian Universities Quality Agency
BC - Business Case
BPA - Business Process Automation
BPEL - Business Process Execution Language
BPEL4WS - BPEL for Web Services
BPEM - Business Process Evolution Management
BPM - Business Process Management
BPMI - Business Process Management Initiative
BPMN - Business Process Modelling Notations
BPR - Business Process Redesign/Re-engineering
CAAC - Courses Approvals and Articulation Committee
CAD - Constraints, Associations, and Dependencies
CBEADS - Component Based E-Application Development/Deployment System
CDMO - Course and Data Management Officer
CEAPC - College Education Assessment and Progression Committee
CGI - Common Gateway Interface
CNI - Course Notification
CSS - Cascading Style Sheets
DAG - Directed Acyclic Graphs
DBMS - Database Management System
DEST - Department of Education, Science, and Training
DISN impacts - Direct, Indirect, Secondary and Non-Cautionary impacts
DWM - Dynamic Workflow Management
EAC - External Advisory Committee
ECA - Event Condition and Action
EDU - Education Development Unit
EPC - Event-driven Process Chains
ERP - Enterprise Resource Planning
FAQ - Frequently Asked Questions
FCP - Full Course Proposal
HEW - Higher Education Workplace
HOS - Head of School
HTML - Hypertext Mark-up Language
IF - Internal Forum
IS - Information Systems
ITD - Information Technology Directory
JEPS - Joint Evolution of business Processes and software System
KAT - Kleene Algebra with Tests
Abstract

CHANGE IMPACT ANALYSIS TO MANAGE PROCESS EVOLUTION IN WEB WORKFLOWS

Organisations have processes to manage their business activities, often referred to as business processes. In today’s competitive global economy, automation of processes with appropriate technology is advantageous. However, the paradox of processes automation is the continuous evolution and change that occurs in business processes. As the business processes evolve and change, the underpinning automated systems need to reflect those changes. Even after a decade of research in the areas of business process automation (BPA) and business process evolution management (BPEM), organisations still find it challenging to manage evolution of automated processes. Therefore, this thesis finds answers to the question of “How can business process evolutions be accurately and effectively reflected in already implemented web-based workflow systems?”

In order to provide a holistic solution to the above research question, this research introduces a framework named paradigm of process automation – PoPA framework and discusses its role in managing process evolution. This framework embodies a business process at four levels as pragmatic, semantic, syntactic, and implementation. Each of these levels deals with a distinctive representation of a business process. For example, the pragmatic level represents the contextual artefact elements such as Acts, policies, organisational structures, rules, and guidelines; that define a process, and the syntactic level denotes the models created for the purposes of automation.

When a change takes place in any one of the levels of the PoPA framework, it creates a propagating impact on elements in the above-mentioned four levels. This propagation of impact takes place due to constraints, associations, dependencies (CAD) among elements within and across the levels (intra and inter-level CAD).

When analysing intra and inter-level CAD most correlations are found to be hierarchical; therefore, a relational database structure is appropriate to capture these hierarchical associations. However, operational processes at the semantic level have
complex associations, which are not hierarchical. Therefore, this research proposes to use Kleene Algebra with Test (KAT) for representing CAD at the semantic level.

Propagating impact does not exclusively depend on inter and intra-level CAD, but is also closely associated with the nature of evolution. Depending on the nature of evolution, the propagating impact can be categorised as direct, indirect, secondary, and non-cautionary (DISN) impact. These DISN impacts suggest the severity of the propagating impact.

The core contribution of this research is the Process Evolution and Change Impact Analysis (PECIA) Model, which enables the management of process evolution accurately and effectively in automated systems.

In this research, a process automation project named Online Courses Approval System (OCAS) is used as an exploratory case study. The practical utility of the PECIA Model is validated using evolution scenarios of OCAS and epistemic utility is analysed based on a study of the literature.

Amidst a plethora of literature on BPA and BPEM, this research is significant due to the following theoretical contributions that facilitate in managing automated processes in tandem with organisational process evolution:

- **PECIA Model** holistically captures inter and intra-level CAD of process elements facilitating the propagating impact analysis within and across the four levels of the PoPA framework.

- A novel use of KAT to capture CAD among process elements cohesively and completely into linear expressions, in order to analyse the impact propagation.

- An algorithm that analyses KAT expressions of a process, to locate DISN-impacts so that evolutions can be carried out accurately and effectively.

The future works that arise from this work are manifold. These may include improving the use of the PECIA Model as a corporate process knowledge repository, and exploring possible other uses of the PECIA Model and KAT based process expressions.
List of Published Work

Elements of this thesis have previously been published elsewhere. The relevant referred publications are:


Chapter 1

"The greatest challenge to any thinker is stating the problem in a way that will allow a solution.”
Bertrand Russell (1872-1970)

1 Introduction

Organisations have processes to manage their business activities, generally referred to as business processes. Advances in information and communication technologies contribute to the partial or full automation of these business processes; and such systems are termed as workflows management systems - WfMS (WfMC, Feb 1999). However, the issue of process automation is the evolving nature of operational processes. When business processes evolve and change the underpinning WfMS need to be modified accordingly.

This thesis presents an approach to solve the problem of accurately and consistently reflecting business process changes and evolutions in web-based workflow systems. This approach is based on cohesively and completely capturing meta-information related business processes and process automation into an analytical model. This meta-information represented in the analytical model assists in locating the propagating impact of process evolution. The identification of propagating impact prior to initiating the modification task in web-based workflow systems facilitates the efficient execution of the task. This in return reduces the downtime of workflow systems that support critical business processes, thus reducing the costs and effort required to manage business process evolution.

Before embarking on detailed discussions, it is important to understand the usage of the two terms ‘evolution’ and ‘change’. Among the handful of researchers who have attempted to differentiate between ‘change’ and ‘evolution’, one clear definition has emerged from the area of managing security processes (George, 2002).
According to Peter J. George (2002), the idiom ‘evolution’ refers to the continuous adaptation of minor modifications that is generally slow and smooth. In contrast, ‘change’ is a course of transformation or conversion, usually responding to a major environmental change, that is generally fast, frequently unexpected and involves some kind of pressure.

Business processes ‘evolve’ and ‘change’ due to various external pressures and organisational needs (these reasons are discussed in detail later in this Chapter). The evolution or change of a business process ultimately results in altering process elements (a more rigorous definition of process elements is defined in subsequent sections) and reflecting them in automated workflow systems. Therefore, a rapid change and slow evolution has the same impact on the process in the context of this thesis. Thus, in this thesis the two terms ‘evolution’ and ‘change’ are used interchangeably.

The rest of this chapter is organised as follows. Initially the primary motivation of this research is discussed, in relation to a practical problem. Followed by the motivation, the research framework that is used to define and organise this research is presented. Based on this research framework, the specific research questions that this study intends to answer are discussed. Subsequently this research is placed into a broader theoretical framework, which identifies other closely related research areas. Next, the contributions and significance of addressing the aforementioned research questions are examined. Finally, this chapter presents an overview of the thesis structure in answering the identified research problem.

### 1.1 Motivation

This research was primarily motivated by the problems associated with a process automation experience, carried out in a tertiary education institute in Australia. This experience relates to the automation of the courses approval processes at the University of Western Sydney (UWS), in a project named OCAS – Online Course Approval System (OCAS, 2006). The processes automated in OCAS are relatively large and complex, which had evolved over six years. Therefore, OCAS is implemented as a web-based workflow system with a number of features to facilitate the anticipated evolutionary nature of the processes. Details regarding
OCAS are presented in Chapter-4 of this thesis, as it is used as the core exploratory case study of this work.

During and after the implementation of OCAS the courses approval processes of UWS continued to change. These changes were predominantly triggered by organisation-wide changes that took place (and are continuing) at UWS. These organisational changes affected the newly automated OCAS in two ways:

(a) The effects of the changes in organisational policies and structures had to be reflected in the already implemented OCAS.

(b) The process managers who participated in the initial implementation effort were replaced by a new set of staff within a year of implementation. This lead to the loss of meta-information in relation to courses approval processes automation in OCAS, which was carried by the initial set of process managers.

The primary focus of OCAS was its flexibility to support the evolutionary nature of the process. Therefore, handling changes such as organisational policies and structures (the above-mention issues labeled (a)), was theoretically achievable.

However the staff turn-over posed a new set of challenges. The inexperience of the new staff in the processes, as well as in OCAS, made them reluctant to use so-called ‘flexibility features’ within OCAS. Therefore, the business analysts of the initial development team were called upon to verify the accuracy of their changes, with questions such as:

“If I move this Dean’s approval two steps down will it create problems?” or

“There is a change in the organisational structure and we do not have a course officer role any more, what places of the process models are affected due to this?”

The above scenario signifies the need for information systems beyond the workflow systems, which capture meta-level information in relation to the process and its automation. Such a meta-information system should facilitate managing changes to:

• organisational policies, procedures, structures, guidelines, laws, and regulations - due to strategic or legal reasons,
• practiced processes due to operational reasons,
• process models due to logical and design errors, and
• implementation artefacts due to technical issues (van der Aalst & Jablonski, 2000).

There is a plethora of research in the area of workflow evolution (a detailed literature review is presented in Chapter-3 of this thesis). In these previous studies, workflow analysis, which refers to process verification, validation, and performance analysis, is identified to be one of the important aspects of managing process evolutions in workflow systems (van der Aalst, 2002; Weske, 2001a). In addition, some researchers (Soffer, 2005; Bodhuin, Esposito, Pacelli, & Tortorella, 2004) have indicated the importance of change impact analysis in managing process evolutions in workflow systems. However, an in-depth and thorough analysis of available literature indicates that there is a significant ‘gap’ in research that attempts to address the issue of change impact analysis. Most research studies, address the problem of workflow evolution from different perspectives, but only provide ‘islands of solutions’.

This view is further endorsed by a recent report by the Gartner group. This report identifies the need of workflow management systems to consist of tools such as repositories, which contain information about process definitions, process components, process models, business rules, and other process data, that would support business process management (Hill, Sinur, Flint, & Melenovsky, 2006). As such, this research aims to provide a holistic solution in relation to impact analysis of process evolution.

In order to comprehend the extent of the above-mentioned problem and to provide a holistic solution, a framework named PoPA - Paradigm of Process Automation is presented.
1.2 Paradigm of Process Automation – PoPA Framework

This framework introduced in this thesis represents the stages or phases associated with process automation; and is adapted from theories and concepts developed by zur Muehlen (2001) and Georgakopoulos, Hornick, and Sheth (1995).

As illustrated in Figure 1.1, the PoPA framework identifies four distinct levels of information representation of the same process, from identification to automation. These levels are: the pragmatic level, the semantic level, the syntactic level, and the implementation level.

In Figure 1.1, each box indicates information representation at a particular level, discussed in detail in subsequent sections. In this figure, the downward arrows represent the flow of information from one level to another, which helps to construct component parts at the target level. The dotted outward arrows represent the ‘leakage’ of information that takes place in the above-mentioned information transformation. The inward arrows illustrate ‘artificial injection’ of information, which does not flow from the main channel. The upward dotted arrows from one
level to another represent influences that lower levels may have on re-shaping the upper levels (Maus, 2001).

1.2.1 Pragmatic Level - Contextual Information

The pragmatic level is the top-most level, which represents different types of contextual information artefacts (Figure 1.1). These contextual information artefacts include laws, regulations, policies, organisational structures, guidelines, goals, values, and strategic plans, which are both internal and external to the organisation.

A business process of an organisation is defined to satisfy a set of internal and external contextual information artefacts (Ramesh, Jain, Nissen, & Xu, 2005; Ali, Soh, & Torabi, 2005b). Therefore, changes to a contextual artefact, such as an organisational structure, have a propagating impact on the business processes. In this view, it is vital to comprehend the association between contextual artefacts at the pragmatic level and the operational business processes represented in the semantic level.

1.2.2 Semantic Level - Operational Business Processes

The semantic level represents the operational business processes of an organisation. These processes consist of elements (Figure 1.1). The literature identifies four types of process elements: actions, participants, object, and rules (Lowenthal, 2003; Hammer & Champy, 1993).

- **Actions** - define the work items or tasks associated with a process.
- **Participants** - refer to computerised agents, software systems, machinery, or humans identified in an organisational role, that perform the process actions. Some literature (Russell, van der Aalst, ter Hofstede, & Edmond, 2005b) refers to participants as resources.
- **Business Object** - refers to an entity upon which the process actions are performed. For instance, a business object could be an informational object such as a ‘leave form’ in a ‘leave approval process’ or a physical object such as ‘car’ in a ‘car manufacturing process’.
- **Rules** - refer to a subset of business rules of the organisation, which facilitates to characterise the above elements and the associations among
them (Hay & Healy, 1997). For instance, ‘employee’ can ‘fill in the leave form’, is a rule that defines the association between the action – ‘fill in the leave form’ and participant – ‘employee’.

Every process element introduced in a business process complies with one or more contextual information artefact identified in the pragmatic level. There is a rationale behind introducing every process element, particularly process actions and associated participants. Process elements are introduced for three purposes: value adding purpose, gate-keeping purpose and communication purpose.

- **Value adding purposes** – Each process has a goal. For instance, ‘approve a leave form’ or ‘enrol a student’. In order to achieve this end goal, certain data needs to be entered, checked, and verified, in process actions. The actions and participants introduced to perform such tasks are present due to value adding reasons.

- **Gate keeping purposes** – In administrative processes, there are certain process actions that are performed as gate keeping tasks. For example in a ‘leave approval process’, the ‘manager’ performs a gate-keeping task, to decide whether the leave form should proceed or prematurely stop.

- **Communicating purposes** – There are certain other actions performed for the sole reason of informing another party or a process. For example in a ‘leave approval process’ the leave-handling clerk will be informed of the final decision of the leave approval process. In some processes this communicative actions may trigger another process to start. In most processes, these types of actions of processes could be fully automated, such as sending e-mails or triggering other processes.

When business processes are defined according to contextual information, there is at least one of the above-mentioned reasons, behind introducing each process element (and its association with other process elements). However, the rationale for introducing process elements (or their associations) is not recorded in any format while defining the process elements. This inability to record the rationale is indicated as ‘information leakage’ in the flow of information from the pragmatic level down to the semantic level in Figure 1.1.
Although the rationales behind process elements are subjected to ‘information leakage’, the processes at the semantic level are enriched due to the tacit knowledge brought in by the process participants. As the processes are practised, different participants bring in different perspectives, based on their experience, training, and cultures (Lewis, Young, Mathiassen, Rai, & Welke, 2007; Weick, 1979). In some situations, this tacit knowledge helps to fine-tune the contextual artefacts in the pragmatic level (Maus, 2001).

1.2.3 Syntactic Level - Business Process Models

The syntactic level denotes visualisations of processes into different models (Figure 1.1). According to McCormac and Rauseo (2005) operational business processes are ‘invisible’. However there is a common understanding among process participants on ‘how things are done’, by development and acceptance of mental models, or deeply ingrained assumptions, generalisations or pictures that provide an understanding of the way things are done (Senge, 1994). The visualisation of processes takes place when processes are re-engineered or automated (McCormac et al., 2005).

Similar to any information systems development, in process modelling three types of process models are created (Whitten & Bentley, 2007b):

- **Process model** – Process model represents various associations that exit within actions, and between actions and participants. For example, a process flow diagram that shows action ordering such as sequence, parallel, and choice, indicates the associations within actions. In addition, it also specifies associations that process participants have with process actions.

- **Data/Object model** – The data model of a process defines the informational objects associated with a process, especially processes that are supported with web-based workflow systems and are associated with informational objects such as ‘leave forms’ or ‘loan applications’. These informational objects are modelled prior to implementation, particularly if these are relatively large, and consist of hundreds of attributes.

- **User interface (UI) definition model** – UI definition models are usually created using informal methods. On some occasions, mechanisms such as
‘story boarding’ are used to model the interface definitions. The main objective of creating UI definitions is to link informational object attributes with action of the process.

Tools used to create the above-mentioned models are mostly developed by information systems experts focused on capturing information vital for automation purposes. These automation-oriented models fail to capture the tacit knowledge that enriches the process. Thus in the information flow from the pragmatic level to the semantic level, the tacit knowledge that fine-tunes the process is leaked out (see Figure 1.1).

1.2.4 Implementation Level – Web-based Workflow Systems

The implementation level of the PoPA framework represents the web-based workflow system (see Figure 1.1) that automates the business process. The concepts of process automation have prevailed over three decades (zur Muehlen, 2004a). During this time, philosophies, technologies, and methodologies associated with process automation have changed dramatically (Baeyens, 2004). The ubiquitous nature and cost effectiveness of the web has made it a popular choice for automating business processes.

As in other information systems, web-based workflows have a number of different components or artefacts such as databases, XML documents, configuration files, function code, style sheets, templates, and web service descriptions (in WSDL – Web Services Definition Language documents). These artefacts are linked together to achieve the desired system functionality. The associations among these implementation artefacts are predominantly decided by the implementation architecture of the web-based workflow system.

Generally, the implementation artefacts are created based on the models developed at the syntactic level. These models have a strong focus on capturing ‘just enough’ information required for the implementation, hence the tacit knowledge that clarifies the processes is omitted in models. However, this type of information is sometimes useful for process participants to interact with the automated systems. For example, when entering data into web forms, certain fields have information points associated, educating the user on the information that needs go into the field.
Therefore, this tacit information is fed back to the implemented systems as help points, tips, or frequently asked questions (FAQs). Since this information is not passed from the main channel, this is referred to as being artificially ‘injected’ into implemented workflow systems.

1.2.5 Role of the PoPA Framework in Process Evolution

The evolutionary journey of elements at each level has a direct mapping to the reasons of evolution introduced by van der Aalst et al. (2000). The contextual artefacts evolve due to strategic and legal reasons, the operational processes change due to operational reasons, the models that conceptualise the process evolve due to logical and design errors and the implementation artefacts change due to technical reasons.

This self-governing nature of each level defines its own journey, and yet their correlations to each other create the problem of propagating impact. For example, when contextual artefacts in the *pragmatic level* change, the processes in the *semantic level* are required to respond to those changes. Similarly, when processes change, the models and implemented workflow artefacts are required to reflect those changes. Therefore, it is important to comprehend the types of associations between the elements of these levels, to manage the propagating impact.

Elements at a particular level, of the *PoPA framework*, do not exist in isolation. For instance, a contextual artefact such as a policy has associations with other policies or regulations. This gives the notion that a change to one policy may have effects on other policies. Similarly, elements in other levels also have correlations among them.

The above discussion identifies that there are constraints, associations and dependencies (CAD) among the elements in each level of the *PoPA framework* as well as across levels. These two types of CAD are termed as:

- **Intra-level CAD** - refers to possible types of correlations among elements within each level (*pragmatic, semantic, syntactic, and implementation*).
- **Inter-level CAD** - denotes the possible correlations between elements across levels such as: between *pragmatic and semantic, semantic and syntactic, and syntactic and implementation*. 
These terms will be used in further discussions presented in the subsequent chapters.

This section presented a framework named Paradigm of Process Automation– PoPA framework, which constitutes four levels: pragmatic, semantic, syntactic, and implementation. In addition, the role of the PoPA framework in process evolutions was discussed. This framework is extensively used in this research presented in the subsequent chapters of this thesis.

1.3 Research Problem

With the understanding of the PoPA framework and its role in process evolution; this section identifies the specific research questions addressed in this research. The main research question addressed is:

*How can business process evolutions be accurately and effectively reflected in already implemented web-based workflow systems?*

In the above research question, there are four key terms: (i) business process, (ii) evolutions and changes, (iii) web-based workflows, and (iv) accurately and effectively. Out of the above list, all terms are previously defined except for the idiom accurately and effectively.

In the context of this research, the management of process evolutions accurately and effectively refers to a holistic approach. The holistic nature of the approach is required to address the following issues:

- The ability to manage evolutions and changes that can initiate at any level (pragmatic, semantic, syntactic, and implementation) of the PoPA framework
- The ability to manage evolutions and changes that take place in all process dimensions (van der Aalst et al., 2000) or elements (actions, participants, and business object)

The solution to the research question is established by breaking the problem into three sub research questions as follows:
Research Question 1: What is the nature of correlations among and across elements of the four levels (pragmatic, semantic, syntactic, and implementation) of the PoPA framework?

This research question is dedicated to understanding both intra and inter-level correlations of the elements in the four levels of the PoPA framework. In answering this question, this research also attempts to develop suitable data structures, and/or formalisms that could be used to conceptualise these correlations in a manner that can be used for analytical purposes. Chapters-5 and 6 collectively contribute to find solutions to this research question.

Research Question 2: Where can changes and evolutions originate and what is the nature (or taxonomy) of process element changes?

This sub research question is aimed at comprehending the evolutionary nature of process elements and its affect on propagating impact. Propagating impact analysis does not exclusively depend on conceptualising both intra and inter-level CAD, but it is factored by the nature of changes in relation to process elements. The nature of process element evolution refers to different types of semantics associated with process element changes. For example, addition (deletion or modification) of a process participant and an action do not have the same semantics. In addition, these distinct changes have different propagating impacts on other elements. For instance, the impact of the deletion of a process participant has different effects on other elements as opposed to deletions of an object data element. For this reason, Chapter-7 is dedicated to comprehending the nature of business process elements.

Research Question 3: Based on the answers to the above two questions; how can the propagating impact of evolutions and changes be identified prior to altering the process models or implemented workflow systems?

The models, concepts, and theories developed in Chapters-5, 6 and 7 are collated in Chapter-8 to answer sub research question 3. In answering this question, Chapter 8 develops the core solution of this research, named the Process Evolution and Change Impact Analysis (PECIA) Model. This model facilitates the analysis of the...
propagating impact that can initiate at any of the four levels of the PoPA framework, down to the implementation artefacts of the web-based workflow system.

In order to find answers to above identified research questions and to verify the applicability of those solutions, a case study will be used in this research. A suitable case study for this research should be of a process automation effort that involves complex and large processes. Secondly, it requires involving processes that changes often, triggering amendments in the automated systems. In addition, the automated system would require to be implemented as a web based workflow system to fit the scope identified in this research. Details of such case study are presented in Chapter-4 of this thesis.

This thesis only uses a single case study, which is significantly large and complex. The rationale for the usage of such a case study is to show the challenging nature of managing process evolutions in large and complex processes, as oppose to hypothetical simple process examples. However, the complex and large nature of the case study hinders the usage of more than one single case study in this research. The inclusion of another large scale case study would move the focus of the key research to the case studies, due to the extensive length of them.

1.4 Context of this Research in the Broader Theoretical Framework

The research focus of this thesis is broadly in the areas of business process automation (BPA) and business process evolution management (BPEM). In particular, this work is focused on finding the impact of process element changes have on already implemented workflow systems. The correlation between BPA, BPEM, and Workflow Evolution (WE) is depicted Figure 1.2. This figure is adapted using the ideas from Kettinger et al.’s (1997) framework for business process re-engineering and Georgakopoulos et al.’s (1995) categorisation of workflows. Figure 1.2 presents the steps involved in BPA and BPEM and it positions the focus of this research in the broader theoretical framework.
The steps in BPA and BPEM are as follows:

- **BPA Step – Process Identification**: This first step in BPA refers to identification of processes for initial automation.

- **BPA Step – Process Re-Design**: Once the processes are identified for automation, these might be redesigned prior to implementation. This step related to BPA is highly researched under synonymic term business process re-engineering (BPR).

- **BPA Step – Create Process Models**: Generally, when processes are selected for automation, a number of models are created prior to the actual automation task. These range from high–level visual models to machine-readable detail models.

- **BPA Step – Model/Workflow Analysis**: Model or workflow analysis refers to the three aspects of validation, verification and performance analysis (van der Aalst, 2002). Validation of process models prior to automating is performed to endorse their conformance with the real-life processes. Verification refers to ensuring that process models are syntactically correct based on the chosen modelling tool. Performance analysis refers to removing bottlenecks for the efficient enactment of the process.

- **BPA Step – Implementation Process Models**: This refers to creating machine-readable process models and setting up initial data, prior to testing by the end-users. Widely used machine-readable process models include XML Process Definition Language – XPDL (WfMC, 2005) and Business Process Execution

- **BPEM Step – Evolution Diagnosis:** This refers to the first step in managing business process evolution. In this step evolution, initiating points would be identified as: *pragmatic level* for strategic reasons and/or *semantic level* for operational reasons and/or *syntactic level* due to design or logical errors and/or *implementation level* due to technical reasons (van der Aalst et al., 2000).

- **BPEM Step – Impact Analysis of Business Process Evolution:** The core focus of this research is in this BPEM step. It refers to the identification of all other component parts affected due to the evolution diagnosed in the previous step.

- **BPEM Step – Process Model Modification:** This refers to modification of actual process models. If the required changes are correctly identified in the previous step, the alteration of models can be carried out in a straightforward manner.

- **BPEM Step – Commit Workflow Changes and Manage Running Instances:** This step denotes reflecting the changes in automated systems and handling the dynamic changes (Casati, Ceri, Pernici, & Pozzi, 1998). The term dynamic changes refer to appropriately managing the running instances of which process definitions have changed (Rinderle, Reichert, & Dadam, 2003; Casati et al., 1998).

- **BPA Step and BPEM Step – Evaluating and Testing:** This is a common step for both initial BPA efforts and subsequent BPEM tasks. This step denotes end-users and/or process managers verifying the accuracy of the implementation or changes carried out.

- **BPA Step and BPEM Step – Rollout:** This is the final step common for both BPA and BPEM. This rollout phase is similar to any system implementation task, where the newly automated or changed system is introduced to a wider user base.

There are a number of methodologies, techniques, and tools (MTTs), that can be used to perform the tasks associated with each of these steps. These MTTs in relation to BPA and BPEM are discussed in detail in Chapter-3.
The focus of this research is to contribute to the body of knowledge of MTTs in relation to impact analysis of business process evolutions. In this view, this research aims to develop a holistic approach to handle evolutions that could initiate at pragmatic, semantic, syntactic, and implementation levels of the PoPA framework in an effective and efficient manner.

1.5 Contributions and Significance of this Research

Having identified the positioning of this work in the broader theoretical framework, this section initially outlines the major contributions of this work. Followed by the contributions, the significance of this research is discussed.

1.5.1 Major Contributions

This research brings a number of valuable contributions to the theoretical body of knowledge of both BPA and BPEM. These main contributions include:

- **Use of Kleene Algebra with Tests (KAT) to capture CAD among process elements:** When capturing the correlations among process elements (actions, participants, and business objects), the requirement is to capture those into an analytical data structure. However, due to the complexity of these correlations (discussed in detail in Chapter-5), it is not possible to represent these in a relational data structure. Therefore, this research proposes to use a formalism named Kleene Algebra with Tests (KAT), which allows the capturing of all associations among process elements into a compact linear expression that can be analysed using an appropriate algorithm (The rationale behind selecting KAT over other formalisms and its usage to capture CAD among process elements are outlined in Chapter-6). The use of KAT to map all possible correlations of a process cohesively and completely, into a single analytical expression, is considered the first contribution of this research.

- **Process Evolution and Change Impact Analysis (PECIA) Model:** This model is the core tool proposed to capture both intra and inter-level correlations into a single cohesive framework. The **PECIA Model** captures the KAT-based process expressions, represented in binary-tree structure (Chapter-6 presents a mechanism and set of invariants to facilitate capturing
KAT expressions into a binary-tree structure), and other associations captured in relational data models. By analysing this model appropriately, the impact propagation of evolutions can be traced to implementation artefacts. The **PECIA Model**, which is a knowledge repository that captures meta-information about process automation, is the second major contribution of this work.

- **Analytical algorithm that locates DISN (direct, indirect, secondary, and non-cautionary) impacts of process evolutions using linear KAT expressions**: This study distinguishes four types of impacts on other process elements as *direct*, *indirect*, *secondary*, and *non-cautionary* (details in relation to DISN impacts discussed in Chapter-7). These DISN impacts are located based on a number of factors, such as the nature of the changes and the position of the changing element in the process. In order to locate DISN impacts an analytical algorithm is developed that can search KAT expressions captured in binary-tree structures. Along with the **PECIA Model**, there is a prototypic implementation of this analytical algorithm, which is the third contribution of this work.

### 1.5.2 Significance of this work in BPEM

Business process management is a topic that has been studied for over a decade to address various problems. Although there are plethora of studies in this area, organisations still find it challenging to reflect process changes in web-based workflow systems successfully.

This view is further endorsed by other researchers and practitioners as follows:

“… *There is a wide gap between the current understanding of BPM by business practitioners on one side, and technology developers on the other side. Unless this problem is addressed, developers will continue to provide business process support solutions that may not bring the expected business value to their adopter.*” (Marjanovic, 2006, p1)

“*Today’s (workflow) systems typically are too rigid, thus forcing people to work around the system. One of the problems is that software*
developers and computer scientists are typically inspired by processes inside a computer system rather than processes outside a computer.”

(van der Aalst, 2002, p5)

In the author’s opinion, this ‘gap’ between business needs and automated systems is further widened by the lack of research that combines operational business issues and technical issues in a single study. Most workflow research related to BPEM concentrates of theoretical issues such as addressing the syntactical correctness of process models, rather than addressing wider issues related to business aspects.

Previous research studies inadequately consider the people factor associated with BPEM. The changes introduced to the system are generally handled by humans in roles such as a business analyst or process manager. When processes are complex and large, keeping track of meta-information in relation to process automation can be an arduous task. This meta-information of process automation relates to the knowledge of associations, such as contextual artefacts’ links with the process elements, process action associations with database tables or attributes, or association among models. At present, this type of information is not managed in any systematic manner, but held by humans who were involved in the initial automation, and/or some are sporadically reported in project documentation.

Business analysts or process managers, not having direct access to meta-information related to process automation, can create errors and inconsistencies in automated systems in BPEM. This problem can be further exacerbated when people move and responsibilities of BPEM change hands. For this reason, many (Marjanovic, 2006; Hill et al., 2006; Ramesh et al., 2005; Ali, Soh, & Torabi, 2005a; Ramesh, 2002) identify the need for knowledge repositories that can record this meta-information in relation to process automation, which allows BPEM tasks to be carried out efficiently.

In an organisational context, knowledge is defined as information that is combined with experience, context, interpretation, and reflection (Davenport, De Long, & Beers, 1997). When this information is collected into a single location for data mining and analytical purposes it is considered to be a knowledge repository (Yoshioka, Herman, Yates, & Orlikowski, 2001).
While there are some research works (Ramesh et al., 2005; Soffer, 2005; Bodhuin et al., 2004) that suggest the use of knowledge repositories for process impact analysis, these solutions fail to provide a holistic approach that spans from pragmatics (contextual artefacts) to implementation artefacts (web workflow component parts). In this view, this research acknowledges the evolutionary nature of contextual artefacts down to the implementation artefacts. Furthermore, it provides a single framework and model to handle these changes, while preserving the integrity of the automated system.

1.6 Thesis Structure

This thesis consists of nine chapters in total, including this introductory chapter. Figure 1.3, which shows the chapter organisation of this thesis. In this figure, the star (*) signs indicate the number of already referred articles published in relation to the work presented in each chapter. Details of the publications are presented on page xiv of this thesis.

Below is a brief overview on Chapters-2 to 9, outlined in Figure 1.3.

- **Chapter 2 – Research Methodology:** This chapter outlines the research philosophy, approach/strategy, methods, and data collection techniques, deployed in this research.

- **Chapter 3 – Literature Review:** This chapter presents out an extensive and thorough review of previous works in relation to BPA and BPEM.

- **Chapter 4 – Exploratory Case Study- OCAS Project:** This chapter describes the *pragmatics, semantics, syntactic* models, and *implementation* details in relation to OCAS project, which is used as the core case study.
Chapter 1 - Introduction

Figure 1.3- Outline of the Chapters in this Thesis

- Chapter 5 – Constraints, Associations, and Dependencies (CAD) of Process Elements: This chapter presents an in depth study in relation to identifying both *intra* and *inter-level* correlations. In addition, relational data structures are developed to capture the CAD of elements at all levels, except for *intra-level CAD* at the *semantic level*. Due to its relative complexity, the discussion of capturing correlations among process elements is presented in Chapter-6.

- Chapter 6 – KAT for Capturing CAD among Process Elements: This chapter justifies the use of KAT as the formalism to conceptualise correlations among process elements. In addition, it also introduces a set of invariants for representing KAT expressions in binary-tree structures.

- Chapter 7 – Nature of Process Element Evolution: This chapter explores the nature of process evolution under, *primitives, semantics*, and *dynamics* of process evolution. Based on the findings, a mechanism is demonstrated to locate the propagating DISN (*direct, indirect, secondary*, and *non-cautionary*) impacts.

- Chapter 8 – PECIA Model: This chapter brings together all the findings from previous Chapters-5, 6, and 7 into a single model (PECIA Model). In
addition, it presents a methodology to use the PECIA Model, in a manner that it can systematically locate propagating impact at the web artefact level. Further, it validates the practical utility of the PECIA Model, using some evolution scenarios drawn from OCAS exploratory case study.

- **Chapter 9 – Discussion, Conclusion, and Future Research Directions:** This final chapter discusses the practical and epistemic utility of the contributions of this research and identifies possible future research directions.

### 1.7 Chapter Summary

This introductory chapter sets the foundation for other chapters, which report the details of this research.

Initially this chapter outlined an issue in relation to a BPEM effort, in the OCAS project as the motivation for this research. This particular issue in the OCAS project highlights the need for a mechanism that manages the meta-information of process automation.

In order to find a holistic solution to the problem, this chapter presented a framework named the Paradigm of Process Automation – PoPA framework. This framework has four levels, identified as pragmatic, semantic, syntactic, and implementation.

The main research question that is addressed in this work is; “**how can business process evolutions and changes be accurately and effectively managed in web-based workflow systems that support them?**” To answer this question, the divide-and-conquer approach is used by formulating three sub research questions. The first sub research question aims to understand two types of correlations, named intra and inter-level CAD, in relation to the PoPA framework. The second sub research question aims to explore the nature of process element evolutions. The final and third sub research question focuses on combining the findings of the first two questions into a single analytical model.

Next, the theoretical framework of this research was presented in relation to two closely associated areas, BPA and BPEM. Then, this chapter outlined the contributions and identified the significance of this research with regards to the plethora of works in BPEM. Finally, the chapter provided an overview of the
structure of this thesis, highlighting the previously published work, listed on page xiv of this thesis.

Having identified the problem and gained an understanding of what the solution should resemble, a mechanism to arrive at the solution is required. The next chapter discusses the research methodology used in this research, to arrive at the solution.
2 Research Methodology

2.1 Chapter Overview

Research in Information Systems (IS) generally combines studies from multidisciplinary areas. As a result, the selection of appropriate research methodology for IS research can be challenging. Consequently, it may require combining multiple research philosophies, approaches, strategies, and data collection methods in a single study. Against such a backdrop, this chapter details the selection process of appropriate research methodologies for the research presented in this thesis.

As indicated in Chapter-1, the research presented is in the broader areas of Business Process Automation (BPA) and Business Process Evolution Management (BPEM). The specific research question, for which answers are sought, is: How can business process evolutions be accurately and effectively reflected in already implemented web-based workflow systems? To answer this question, the divide-and-conquer approach is used, by formulating three sub-research questions as follows:

- What is the nature of correlations among and across elements of the four levels (pragmatic, semantic, syntactic, and implementation) of the paradigm of process automation - PoPA framework?
- Where can changes and evolutions originate and what is the nature (or taxonomy) of process element changes?
• Based on the answers to the above two questions, how can the propagating impact of evolutions and changes be identified prior to altering the process models or implemented workflow systems?

In this chapter, the characteristics of this research are first examined to decide the appropriate research methodology. The selection process involves deciding the broader research philosophy, based on the philosophical view of the researcher. The research philosophies considered include positivism, interpretivism, and critical research (Orlikowski & Baroudi, 1991). By analysing the characteristics associated with qualitative, quantitative, inductive, and deductive research methods and mapping them against the attributes of this research, the appropriate research approach for this work is determined. There are a number of research techniques that can be used for IS and business-related research work. These techniques include case study, experimental, survey, action, and constructive methods. By analysing the features associated with each of these methods and their appropriateness in answering the above sub-research questions, a set of suitable methods are selected to use in this research. Finally, the data collection methods suitable for this research are decided. The above-mentioned criteria, which includes research philosophies, approach, techniques and data collection methods, is based on the research framework suggested by Saunders, Lewis, and Thornhill (2003).

The rest of this chapter is organised as follows. Firstly, Saunders et al.’s (2003) ‘research process onion’, which sets the research framework, is presented. Following this introduction, the selection process according to the above criteria is explained. Finally, the applicability of the selected approaches, techniques, and methods, to various parts of this research is indicated in the diagrammatic format.

2.2 The Research Framework

The concept of the ‘research process onion’ (Figure 2.1) by Saunders et al. (2003) gives a practical framework to carry out the selection of the methodology associated with any research. The word ‘onion’ gives the impression of having layers that encapsulate inner layers – by peeling away the outer layers we discover what resides underneath. In this view, there are five layers identified in the research
process that set the criteria for selecting an appropriate methodology for a research work.

![Figure 2.1- Layers in Saunders et al.'s (2003) Research Process Onion](image)

The layers identified in the ‘research process onion’ are Research Philosophy, Research Approach, Research Strategy, Time Horizons, and Method of Data Collection. These layers are explored below to identify the content that falls within each layer:

- **Research Philosophy**: The broader research philosophy gives an identity to a research work. The philosophy associated with a research study is based on the philosophical view of the researcher. There are a number of philosophical studies such as Positivism, Interpretivism, Negativism, Historicism, Critical Rationalism, Classical Hermeneutics and Realism, used in modern research (Blaikie, 1993). From this list, Positivism, Interpretivism, and Critical Rationalism are further explored to determine their applicability to this research. The basis for this selection is their wider usage in previous IS and business-related research work (Orlikowski et al., 1991).

- **Research Approach**: The research approach requires justifying two aspects related to a particular research work. These two aspects are; i) **qualitative** or **quantitative** nature of the research, and ii) whether the research is **deducted** from any existing theories to justify specific scenarios or **inducts** new theoretical concepts (Saunders et al., 2003). Characteristics of this research are
explored to determine their qualitative and quantitative nature, and the relevance of deduction and induction approaches for this work.

- **Research Strategy:** The research strategy, also known as research method (Orlikowski et al., 1991) requires the suitability of certain techniques to conduct the research to be considered. From this perspective, a number of methods including Survey, Lab Experiment, Exploratory Case Study, Field Experiment, Instrument Development (Constructive Research), Design Research, and Action Research (Kasanen, Lukha, & Siitonen, 1993; Orlikowski et al., 1991) are closely analysed to determine their applicability in achieving the objectives of this research.

- **Time Horizon:** Cross-sectional and longitudinal are the suggested time horizons that define the period of study related to a research undertaking (Saunders et al., 2003; Orlikowski et al., 1991). The term cross-sectional refers to studies that collect data in a relatively short period. Generally, in cross-sectional studies data is not collected repetitively. Conversely, in longitudinal studies data is collected repetitively over a long period (Orlikowski et al., 1991). Therefore, without further analysis, it can be concluded that this research is a cross-sectional study, as data is not repetitively collected over a long period. Consequently, in the next section, the time horizon criterion will not be examined further.

- **Method of Data Collection:** There are number of data collection methods deployed in IS research. These data collection methods include sampling, questionnaires, observations, interviews, documentation, and secondary data from methods such as literature reviews (Saunders et al., 2003). As more than one data collection method is applicable to this research, further analysis will be carried out in the next section to find the appropriate data collection method for this research.

  The next section explains the selection process of the research methodology for the work detailed in this thesis.
2.3 Selection Process of the Research Methodology

In IS research, it requires to combine more than one research approach or methodology to achieve the desired outcomes (Chinh, 1995). This section details the selection processes of a blended methodology for this work.

Prior to discussing the methodology selection criteria, it is appropriate to list the characteristics and goals associated with this research, as follows:

- In this research, a definite research problem is identified in the form of a research question (and sub-questions) and does not involve a hypothesis.

- A solution is devised to answer this identified research problem by conceptualising and systematically modelling the associations between various contextual artefacts (Laws, Policies, Regulations and Organisational Structures), operational business process elements (actions, participants and business object), syntactical models (process models, data models and interface models) created in automation, and implementation artefacts (databases, code files, documents, styling templates and frameworks).

- The applicability of the solution must be verified to prove it answers the research question(s). In addition, it is required to show the repetitive applicability of solutions to solve similar problems in BPEM. Further, it is necessary to distinguish the new knowledge discovered in this research, in the process of answering the initially set research question.

The characteristics of this research highlighted above will be considered, when deciding the suitable research methodology for this work.

2.3.1 Research Philosophy

There are three main philosophical perspectives (positivism, interpretivism and critical research) that are applicable to IS research (Myers & Avison, 1997; Orlikowski et al., 1991).

According to Orlikowski and Baroudi (1991, p10) positivist IS researchers assume “an objective physical and social world that exists independent of humans, and whose nature can be relatively un-problematically apprehended, characterised,
and measured”. Using this view, the researcher believes that the object of a study can be modelled to capture its behaviour, to measure, and analyse.

In contrast to positivism, the interpretivism “asserts that reality, as well as our knowledge thereof, are social products and hence incapable of being understood independently of the social actors (including the researcher) that construct and make sense of that reality” (Orlikowski et al., 1991, p14). In this view, the researcher actively participates and tries to understand subjective meanings embedded in the object of study. In Walsham’s (1995) view use of interpretive methods in IS are aimed at producing an understanding of the environment in which the IS exist, and the process whereby the IS influences and is influenced by the environment.

The critical philosophy aims to overcome some of the problems identified by interpretivism, such as adding an evaluation dimension that allows the critiquing of situations, considering conditional aspects associated with situations and taking the historical influences on the object of study (Orlikowski et al., 1991). Another important aspect associated with critical research and closely relates to the author’s view is the philosophy of totality. In other words, “particular elements exist only in the context of the totality of relationships of which it is a part, and this part and the whole are bound by mutual interaction” (Orlikowski et al., 1991, p21).

The three philosophical approaches (positivism, interpretivism and critical research) and their applicability to this research is analysed in Table 2.1. In Table 2.1, meta-theoretical assumptions on ontology, epistemology, theory of truth, research object, method, validity, and reliability are used to compare the three philosophies. Furthermore, this comparison is used to position this research in an appropriate philosophical paradigm. This table is based on the initial works of Weber (2004) and extended to include the critical research paradigm.
Table 2.1 - Meta-Theoretical Assumptions in Positivism, Interpretivism and Critical Research - Adapted based on Weber (2004) and Orlikowski et al. (1991)

<table>
<thead>
<tr>
<th>Meta-theoretical assumption on (Weber, 2004)</th>
<th>View of the author and/or the characteristics of this research</th>
<th>Positivism</th>
<th>Interpretivism</th>
<th>Critical Research</th>
<th>Use in this research</th>
<th>Use in this research</th>
<th>Use in this research</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ontology:</strong> is the study that describes the nature of reality</td>
<td>Reality is that business process evolutions and its propagating impact takes place independent of the researcher, models and implementations</td>
<td>Person (researcher) and reality are separate.</td>
<td>✓</td>
<td>Person (researcher) and reality are inseparable (life-world).</td>
<td>X</td>
<td>Reality is historically constituted, and hence not confined to a particular state</td>
<td>X</td>
</tr>
<tr>
<td><strong>Epistemology:</strong> is the study that explores the nature of knowledge</td>
<td>Knowledge on nature of business process evolutions is there to be discovered by the researcher</td>
<td>Knowledge exists beyond the human mind to be discovered</td>
<td>✓</td>
<td>Knowledge of the world is intentionally constituted through a person’s lived experience.</td>
<td>X</td>
<td>Knowledge is grounded in social and historical practices.</td>
<td>X</td>
</tr>
<tr>
<td><strong>Theory of Truth</strong></td>
<td>Focus is on uncovering the knowledge creatively constructing and controlling the environment. This involves model creation to solve a problem, rather than for proving (or disproving) a hypothesis</td>
<td>Correspondence theory of truth: one-to-one mapping between research statements and reality</td>
<td>X</td>
<td>Truth as intentional fulfillment: interpretations of research object match lived experience of object.</td>
<td>X</td>
<td>No theory and interpretation conclusively prove or disprove a theory. Status of anything is merely one moment and things evolve continuously</td>
<td>✓</td>
</tr>
<tr>
<td>Meta-theoretical assumption on (Weber, 2004)</td>
<td>View of the author and/or the characteristics of this research</td>
<td>Positivism</td>
<td>Interpretivism</td>
<td>Critical Research</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---------------------------------------------------------------</td>
<td>-------------</td>
<td>----------------</td>
<td>------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Research Object</strong></td>
<td>Research object (business process evolutions) exists independent of the researcher. However, the process elements do not exist in isolation; hence create propagating impact in evolution. Further object can be modeled for analysis.</td>
<td>Use in this research</td>
<td>Interpretivist’s view</td>
<td>Critical Researcher’s view</td>
<td>Use in this research</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Research object has inherent qualities that exist independently of the researcher.</td>
<td>✓</td>
<td>Research object is interpreted in light of meaning structure of person’s lived experience.</td>
<td>X</td>
<td>Research object cannot be treated as isolated elements and bound by mutual interactions with other objects in the environment</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td><strong>Methods</strong></td>
<td>Case Study, Sample evolution scenarios and Secondary data from literature</td>
<td>Use in this research</td>
<td>Interpretivist’s view</td>
<td>Critical Researcher’s view</td>
<td>Use in this research</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Samples, devising, controlled experiments, Statistical analysis</td>
<td>✓</td>
<td>Field studies, observations, interviews</td>
<td>X</td>
<td>Long-term historical studies, and ethnographic studies</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td><strong>Validity</strong></td>
<td>Analytical data captured in model analysis exactly indicate the nature of evolution of processes</td>
<td>Use in this research</td>
<td>Interpretivist’s view</td>
<td>Critical Researcher’s view</td>
<td>Use in this research</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Certainty: data truly measures reality</td>
<td>✓</td>
<td>Defensible knowledge claims</td>
<td>X</td>
<td>Evolving and emergent dynamics of the reality</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td><strong>Reliability</strong></td>
<td>Under similar circumstances (similar process) the results can be reproduced or repeatedly applied for other business processes</td>
<td>Use in this research</td>
<td>Interpretivist’s view</td>
<td>Critical Researcher’s view</td>
<td>Use in this research</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Repeatability: research results can be reproduced.</td>
<td>✓</td>
<td>Interpretive awareness: researchers recognise and address implications of their subjectivity.</td>
<td>X</td>
<td>Results cannot be re-produced easily due context changes that take place in the object</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Legend – (✓) Appropriate and agrees with this characteristics of this research and (X) Inapplicable and does not comply with features of this research
According to the findings shown in Table 2.1 this research has more inclination towards the positivist paradigm. However, this research does not involve a hypothesis but a research question (under the meta-theoretical assumptions about the ‘theory of truth’); this is the only reason that precludes it from being in the positivist perspective. The use of a research question instead of a hypothesis suggests the possibility that this research belongs to interpretivist and critical research.

This associative characteristic is a fundamental principle on which this research is built. This refers to the belief that its associations with other elements in the environment define the very existence of an element. This is another reason to argue that this research belongs to critical research.

Considering the information given in Table 2.1 and based on the above arguments, the conclusion is that this research, similar to most IS-related research studies (Orlikowski et al., 1991), can be classified as positivist research, with certain characteristics from interpretivism and critical research. The main reasons for it to be in the positivist paradigm is the objective view held by the author in modelling the associations among process elements and evolutions. Secondly, these models are used for analytical purposes, where due to its objective nature; the research data can be reproduced and applied repeatedly. Although it has certain elements that suggest this work to be in interpretivist or critical research paradigm, these elements are not sufficiently strong enough to preclude it from the positivist perspective.

### 2.3.2 Research Approach

According to Saunders et al.’s (2003) research process framework, the research approach requires a decision between qualitative or quantitative nature, and inductive or deductive aspects of a research study.

Quantitative research method uses numerical analysis such as survey methods, laboratory experiments, formal methods, and numerical methods such as mathematical modelling (Myers et al., 1997). In contrast, qualitative research involves the use of qualitative data, such as interviews, documents and participant observation data to understand and explain a phenomena (Myers et al., 1997).

In this research, the primary method of data collection is based on scenarios drawn from an exploratory case study (Yin, 2003; Yin, 2002). The definitions of qualitative and quantitative approaches given by Myers and Avison (1997) fail to
specify the exact nature of case study research. In addition, King and Applegate (1997) believe that it is not possible to separate the qualitative and quantitative nature of a case study, as it may involve both types.

This research also involves a certain element of algebraic modelling, where correlations among process elements are mapped into linear algebraic expressions based on Kleene Algebra with Tests (KAT). While these expressions are used for analytical purposes, it does not produce any numerical results, but a set of process elements that require further evolutions. However, in verification of the model, certain sample scenarios are used that are related to the case study. This approach is similar to sampling, which suggests this research to be quantitative.

Subsequently, it can be concluded that this research uses some elements of quantitative methods to arrive at qualitative results. However, these qualitative results are not as vague or descriptive as in social sciences, but are definite and conclusive with regards to the related business process and evolution scenario.

Induction refers to generalisation or the discovery of new theories; deduction indicates the use of general theories to find answers or understand a specific problem (Blaikie, 1993). From this perspective, this research uses both inductive and deductive approaches (Perry & Jensen, 2001) in combination as follows:

- The solution is deduced from various existing theoretical bodies of knowledge such as process modelling theories, algebraic modelling, workflow-analytical theories, and searching mechanisms. These previous theories are used to comprehend the problem of process evolution and devise a solution for evolution impact analysis.

- While devising the solution, this research also develops a new, yet cohesive and complete process-modelling tool suitable for capturing all possible associations among process elements. This new modelling method creates executable linear expressions of process models using KAT. The discovery of this applicability of KAT for process modelling is considered one part of the new knowledge discovered in this research.

The application of both induction and deduction methods in this research is further detailed in Figure 2.2.
In summary, from the qualitative and quantitative approaches point of view, this research is mainly categorised as a qualitative research, which uses some quantitative methods. In addition, this research uses both inductive and deductive approaches in understanding the problems and devising its solution.

### 2.3.3 Research Strategy

In Saunders et al.’s (2003) ‘research process onion’, the third layer refers to identifying the applicable research strategy or method to solve the research problem. Here the methods such as Survey, Action, Experimental, Case Study, Design (Gregor & Jones, 2004; Hevner, March, Park, & Ram, 2004; March & Smith, 1995) and Constructive (Cornford & Smithson, 1996; Kasanen et al., 1993) are investigated to evaluate their applicability to this work.

In Survey research, a considerable sample selected from a population, to analyse certain characteristics or frequencies of the population (Kock, 2003). Information is gathered using methods such as questionnaires. The main purpose of
the surveying method is to generalise the findings to the whole population. The research presented here does not involve questionnaires or survey methods where significant amounts of sample data are collected. Therefore, the applicability of the survey method to this research can be considered low.

Action research is considered to be of dual purpose, where it aims to improve the situation being studied and generate relevant new knowledge (Kock, 2003; Baskerville, 1996). This method extensively uses the participatory approach by the researcher, where data is collected through observations and interviews. The research detailed in this thesis does not involve any of these data collection methods, hence precluding the applicability of the action research method to this work.

Experimental research is used to investigate a problem under a controlled environment. In the experiments the variables are manipulated, associated numeric data is collected and causal or correlation models are tested through standardised statistical analysis procedures (Kock, 2003). In this research, the correlation model that represents the associations of process elements is tested using some scenarios drawn from the case study. While this testing does not involve rigorous statistical analysis, the testing is carried out in a controlled manner to measure the propagating impact of an element change. Accordingly, there are some experimental aspects associated with this research.

The Case Study method is considered to be one of the most widely used methods in IS research (Myers et al., 1997; Orlikowski et al., 1991). Yin (2003; 2002) says that exploratory case studies allow the researcher to discover problems, understand the nature and complexity of what is happening and to uncover new emerging topic areas for further research. In this research, the exploratory case study based on the Online Course Approval System (OCAS) development project (OCAS, 2006) was the primary motivation for revealing the problem. In addition, the knowledge gained from the case study gave an insight to the in-depth investigation required to solve the problem.

The two other research strategies – design and constructive research methods are discussed together due to some similarities in techniques used. Design research deals with creating something new that does not exist in nature and outputs constructs, models, methods, and instantiations (Gregor et al., 2004; March et al., 1995). The objective of the design method is to solve a problem or understand a phenomenon. Similarly, the constructive research method involves creating models,
systems and software with the philosophical view that the world is articulated through step-by-step construction, from supposedly basic elements like objects, time-pace slices, observations, thoughts, or logical relations (Kasanen et al., 1993).

Both design and constructive research methods have a set of steps referred to as analysing or comprehension of the problem, deducing knowledge from existing theories and bodies of knowledge, development and validation (Hevner et al., 2004; Gregor et al., 2004; Kasanen et al., 1993). The validation of the construct (or the designed artefact) can be done using a multitude of techniques such as; observations (through case study or field study), analysis (through statistical analysis, architecture analysis, optimisation or dynamic analysis), experiments, testing (through functional or structural testing), and descriptions (through informed arguments or scenarios) (Hevner et al., 2004).

The major difference between the design and constructive approaches is based on the objectives of the validation. The early works on design research (March et al., 1995) only validates the suitability of the designed artefact to solve the problem or understand the phenomena. In contrast, the constructive research emphasises the validation of both the practical and epistemic utility of the construct (Kasanen et al., 1993). However it is worth noting that recent works on design method (Gregor et al., 2004; Hevner et al., 2004) embed steps for epistemic evaluations as well.

The core work of this research can be mapped to the steps in the constructive research method, as it is more methodical and appropriate for this research. The steps in constructive research are:

- Detecting and defining a problem;
- Investigating the existing theoretical body of knowledge to deduce that the problem is worth solving;
- Constructing a solution;
- Verifying that the solution to ensure that it solves the initially set problem - practical relevance; and
- Evaluating the newly discovered knowledge in order position it within the existing body of knowledge – epistemic relevance.
Figure 2.3 illustrates the application of the steps in constructive approach in this research.

The research process starts with the problem identification. The problem addressed in this research is to solve the issue of managing process evolutions effectively and efficiently in web-based workflows. In addition, this research attempts to find a holistic solution, which:

a) Allows managing evolution from all process dimensions – actions, participants and object,

b) Facilitates finding the propagating impact from contextual artefacts, such as policies, regulations, laws and organisational structures, down to the implementation artefacts of the web-based workflow systems.

The second step in constructive research explores the theoretical body of knowledge with two objectives. One objective is to confirm that the identified problem is worth exploring and that no one has previously solved this problem. The other objective is to identify relevant theoretical knowledge upon which a solution
can be built on. In this research, the theoretical body of knowledge studied includes the following:

- Business Process Automation (BPA) and Business Process Evolution Management (BPEM)
- Process and Workflow Evolution
- Workflow Analysis and Impact Analysis
- Process Modelling Methods, including Petri-Nets and Process Algebra (Oren & Haller, 2005)
- Kleene Algebra with Test (KAT)
- Data Structures and analytical algorithms

This theoretical knowledge is further enhanced by the fuzzy information that the researcher possesses or has access to. This fuzzy information may include previous training, work experiences, organisational practices, and cultural backgrounds. This fuzzy information is brought into the research by the researcher and/or generated through discussions that the researcher has with other researchers and practitioners.

Solutions that provide answers to the identified problem consist of two parts. Firstly, it has an analytical model that captures correlations of process elements. This model is named the **Process Evolution and Change Impact Analysis (PECIA) Model**. Secondly, the solution consists of an algorithm that is capable of searching the model in order to reveal the propagating impact of process elements.

The practical utility of the solution is verified by using the OCAS-related processes as case studies. Initially, the associations among process elements in OCAS are mapped into the **PECIA Model**. Then, certain selected process evolution scenarios are run through the algorithm to verify that it produces the required results. This validation method has some experimental aspects associated with it, as it is carried out in a controlled environment. For example, while testing the propagating impact of one element (for example, a process participant) evolution, the other dimensions (actions and object data) are considered constant.

The epistemic utility of the solution is confirmed predominantly through literature review. The application of KAT to creating process models introduces
some new theoretical concepts in process modelling, particularly, the use of a bi-partite algebra (KAT) for process modelling, which allows some aspects that were not previously possible using uni-partite algebras (Process Algebra). This epistemic utility is further proven by solving a challenge that was previously set by another researcher (van der Aalst, 2003) to model a particular process scenario, which was not possible using Process Algebra.

In summary, this section evaluated the applicability of a selected set of research methods to this research. Based on this discussion, the main conclusion is that this research uses constructive research as the primary research method. However, in the validation steps of constructive research certain scenarios drawn from case studies and some experimental methods are used. Further methods such as survey and action research are considered unsuitable for use in this research.

### 2.3.4 Methods of Data Collection

The primary methods of data collection used in this research include case study based process evolutions samples and study of the secondary data such as OCAS implementation details, organisational evolutions and literature review.

OCAS, which serves as the primary case study, was developed in-house by University of Western Sydney (UWS) – Australia. The OCAS-related implementation details were thoroughly analysed to gain an understanding of the implementation details of a web-based workflow management systems. Furthermore, the organisational processes (courses approval process in UWS) automated in OCAS were examined in detail to comprehend associations of process elements, not only within the process but also with contextual artefacts (laws, policies, organisational structure), models created in implementation (process models, data models and interface models) and web-implementation artefacts (database, documents and code files). Based on real evolutions that took place during the time of the research, a set of evolution scenarios were selected to test the validity of the solutions.

A vast number of research studies exist in the areas of BPA and BPEM, as they have proven popular areas in the past decade. Therefore, it was very challenging to select the work required to be covered in this research, particularly in a manner that adequately covered a cross-section and landmark work. However, due to features
and searching power of Google Scholar site (http://scholar.google.com), electronic databases including ACM digital library (http://portal.acm.org/portal.cfm), IEEE Xplore (http://ieeexplore.ieee.org/Xplore/guesthome.jsp), Springer-Link (http://www.springerlink.com/home/main.mpx), and CiteSeer Scientific Literature Digital Library (http://citeseer.ist.psu.edu/) it was possible to shortlist the required literature, by analysing previous referencing patterns and associated work.

2.3.5 Level of Applicability of the Selected Methodology

The selection process of the suitable philosophies, research approaches, strategies, and data collection methods for this research study were detailed previously. The applicability of the selected methodology can be summarised into a single model as shown in Figure 2.4.

| Research Philosophy       |  |
|---------------------------|  |
| Positivism                | Critical Research |

| Research Approach         |  |
|---------------------------|  |
| Qualitative (using some quantitative methods) | Deduction followed by Induction |

| Research Strategy/method  |  |
|---------------------------|  |
| Constructive Research     | Case Study | Experiment |

| Time Horizon             |  |
|--------------------------|  |
| Cross Sectional          |  |

| Method of Data collection|  |
|--------------------------|  |
| Secondary data based on Case Study and Literature | Observations | Sampling |

*Figure 2.4 - Summary of the Research Methodology Applicable to this Research*

Figure 2.4 shows the selected philosophies, approaches, and techniques based on the layers in the research process framework presented in section 2.2. In Figure 2.4, each of the selected research philosophies, approaches, methods or strategies and techniques, are represented in varying size boxes. The length of each box gives an approximate indication of the relevance or the usage of that concept in this research. For example, under research strategy, constructive research is predominantly used in comparison to case study and experimental methods.
2.4 Application of the Selected Methodology in this Research

The Constructive research, Case Study, and Experimental methods were selected as the main research strategies to be used in this research. These three methods are used in varying degree in this research. Figure 2.5 illustrates the applicability of these research strategies in the main chapters of this thesis.

![Figure 2.5- Application of Selected Research Strategies in this Thesis](image)

According to Figure 2.5, it is evident that constructive methods are heavily utilised in the core chapters of this research. The case study approach and experimental methods are also used in some chapters as illustrated above.

The case study and experimental methods are used to complement the core constructive research method. For instance, the OCAS-related case study is used to comprehend the problem of workflow evolution in web-based workflows. The problem comprehension is one-step involved in constructive research. Additionally, experimental methods are used to validate the PECIA Model in Chapter-8. Similar to problem comprehension, the validation is another step involved in constructive research.

2.5 Chapter Summary

This chapter detailed the selection process of the research methodology and its application in this thesis. The selection process was based on the framework of the ‘research process onion’ by Saunders et al. (2003).
In the above-mentioned selection process, the characteristics of the research are matched against the features of each research philosophy, approach, method, time horizon, and data collection techniques, in order to select the most appropriate methodology at each level. Based on the selection process, it was ascertained that this research is a positivist research, which has some characteristics of critical rationalism. By considering the quantitative and qualitative aspect, this research has qualitative research goals, but uses some quantitative methods to achieve those goals. This research uses both deduction and induction approaches in this work. The main research strategy used in this work is constructivism. In some steps of the constructive research, case study and experimental methods are used. From the time horizon point of view, this research distinctly belongs to the cross sectional as opposed to longitudinal. Finally, the case study based observations, sampling and secondary data based on literature review was identified as the main data collection techniques in this research.

With this background knowledge of the suitable research methodology, the next chapter carries out an in-depth and extensive literature review. This literature review focuses on the areas of BPA and BPEM.
Chapter-3

“Anybody can make history. Only a great man can write it.”
Oscar Wilde (1854-1900)

“There is nothing like looking, if you want to find something. You certainly usually find something, but it is not always quite the something you were after.”
J.R.R. Tolkien (1892-1973)

3 Literature Review

3.1 Chapter Overview

Business processes and workflow are heavily researched topic areas, in relation to information systems. Some argue that workflow related research and practices date back to the 1970s, when office automation was a widely discussed topic (zur Muehlen, 2004a). Nevertheless, from the point of view of advancement, some believe that workflow and business process management (BPM) have a long journey ahead, in comparison to more established areas such as database management systems (DBMS). With this view, Baeyens (2004) has positioned the workflow and BPM research at a point much earlier than the ‘peak of inflated expectation’ in the hype curve as shown in Figure 3.1.

![Figure 3.1- Workflow Positioned in the Hype-Curve, adopted from Baeyens (2004)](image_url)
Despite much relentless work by numerous researchers and practitioners, it is interesting to understand the reasons for positioning workflow and BPM in the early stages of the hype curve. Giving some insights into this problem, a recent report from Gartner Group\(^1\) (Hill et al., 2006) identifies many sub-topic areas that require further research and development in relation to BPM. From the many such sub-topic areas identified in the above-mentioned report, the following two are closely related to the research presented in this thesis:

- The need for more facilities that business managers can use to control and modify their processes, and
- Synergies between technologies and methodologies to enable business managers to make rapid process improvements.

This view of the Gartner Group is further supported by many researchers who have identified the need for further improvement in various areas of BPM, particularly in relation to managing process evolutions (Marjanovic, 2006; Zur Muehlen, 2004c; van der Aalst, Weskez, & Wirtz, 2003c; Stohr & Zhao, 2001; van der Aalst et al., 2000). This aspect of managing business process evolutions and changes in workflow systems are studied under a number of headings such as workflow evolution, workflow adaptation, dynamic workflows, workflow modifications, and workflow flexibility (Sadiq, Marjanovic, & Orlowska, 2000a).

The research presented in this thesis is broadly in the area of business process evolution management (BPEM) in workflow systems. The emphasis of this work is to explore and develop an approach that allows analysing the impact of business process changes in workflow systems. In other words, this research is focused on exploring method/s that allow business analysts to effectively find the impact of business process changes in already implemented web-based workflow systems, prior to changing the process models or workflow systems. The key advantages of such impact analysis system are; that it reduces the errors human business analysts would make in changing process models. In addition, it supports the evolution task to be carried out by someone who was not originally involved in the automation.

\(^{1}\) Gartner, Inc. (NYSE: IT) is one of the world’s leading information technology research and advisory company for over 10,000 clients that comprise of CIOs and senior IT leaders in corporations and government agencies, business leaders in high-tech and telecom enterprises and professional services firms and technology investors. [http://www.gartner.com/it/about_gartner.jsp](http://www.gartner.com/it/about_gartner.jsp)
This work of business process evolution and change impact analysis is closely related with areas such as *business process automation (BPA)*, *business process evolution management (BPEM)*, *workflow technology*, *process modelling*, *flexibility in workflow systems*, and *workflow analysis*. Hence driven by the objective of positioning the work presented in this thesis in a suitable theoretical framework, this chapter is aimed at presenting the ‘state-of-the-art’ developments in the above-mentioned core and some other peripheral areas. In addition, this chapter presents an in-depth analysis of previous work to investigate possible technical concepts that could be combined together in arriving at the intended solutions, for the problem of managing business process evolution.

This chapter is broadly organised into four sections. The first section clarifies the usage of some key terminologies used in this research, highlighting different interpretations of their usage in other literature. Then the theoretical framework is presented illustrating the positioning of this work and highlighting key methodologies, techniques, and tools (MTTs) in the related areas. Followed by the theoretical framework, related other research areas are discussed to comprehend the current position of research in relation to BPEM. Finally, overview of fundamental technological concepts that have potential to be used in developing the solutions in this research are presented.

### 3.2 Key Terminologies

This section presents a set of terminologies that form the foundation for this research. These terms are *Processes*, *Business Processes*, *Business Process Elements*, *Process Automation*, *Workflow*, *Workflow Management Systems (WfMS)*, *Process Definition and Process Instance*. Definitions of these terms and their usage in the context of this work are first presented here.

#### 3.2.1 Process, Business Process, and Process Elements

The term process has distinctive domain usages. For example, according to The Macquarie Dictionary (2007) a *process* may refer to manipulation of data in order to abstract the required information, while in photography it refers to various photomechanical methods used for developing photographs. In this sense, *process* refers to a systematic series of actions directed to some result.
When the result expected from a process relates to a business goal of an organisation, the term *process* becomes synonymous with *business process*. There are a number of definitions for the term *business process*. One of the commonly used definitions refers to a *business process* as a set of logically related tasks performed to achieve a defined business outcome (WfMC, Feb 1999; Malhota, 1998; Hammer et al., 1993; Davenport, 1993). In the context of this research, the two terms *process* and *business process* are used interchangeably.

The dimensions, elements, or perspectives of a process play an important role in defining a process. When analysing the previous literature it is evident that researchers have used different terms based on the focus of their research.

Researchers such as Lowenthal (2003), Malhota (1998), Stoddard and Jarvenpaa (1995) and Davenport (1993), use the terms *tasks*, *actors*, *documents or products*, and *business rules* to identify the elements of a business process. In workflow research, which is more technically oriented, these four dimensions are referred to as *tasks, resource, information* and *process* (van der Aalst et al., 2003c; van der Aalst, 2002; Weske, 2000). Mangan and Sadiq (2002b) refer to these dimensions as *tasks, performers, objects* and *constraints*.

These different terminologies represent the same concepts in identifying business process elements. In the context of this research, the terms *actions, participants, object*, and *rules* represent the elements of a business process.

- **Actions**: A process may involve a number of work items performed to achieve a business goal. Examples of some actions associated with loan approval processes are filling in the loan application, checking credit rating and final loan approval. The term *action* is similar to the term *task* used by other researchers (Mangan et al., 2002b; van der Aalst, 2002; Weske, 2000).

- **Participants**: Process actions are performed by *participants*, which could be organisational departments, machinery or software applications or human participants (Russell et al., 2005b). In the loan approval example, participants could be a loan applicant, bank clerk and bank manager. In this research the term *participant* is used similar to the terms *resources* by Russell et al. (2005b) and Weske (2000), and *performers* by Mangan et al. (2002b).
• **Business Objects:** Process actions are performed in order to manipulate a business object, which could be physical or informational (Zur Muehlen, 2004c; Georgakopoulos et al., 1995). In the loan approval example, the data associated with the loan application can be represented as an informational business object in automation. A business object of a process is referred to as information perspective by some researchers (Weske, 2000).

• **Rules:** A process has a set of rules or flow rules that define the association between actions, participants, and objects within a period of time. These business rules are not limited to defining the routing rules (example *if the loan amount is $1,000,000 it requires signatures from two bank managers*), but also helps in defining the terms (loan, application, applicant, clerk and manager) and the associations (applicant fills in the application and the clerk checks the credit rating of the applicant) (Hay et al., 1997). This is similar to the terms process perspective (van der Aalst, 2002; Weske, 2000) or constraints (Mangan et al., 2002b) used by others.

While all business processes have the above-mentioned four types of elements or perspectives as some researchers refer to it (Weske, 2000), there can be certain differences among business processes. This leads to a categorisation of business processes based on different viewpoints.

One of the earliest categorisations is as core, support, and management processes (Childe, Maull, & Bennett, 1994). The core processes focus on the activities that bring profit to the organisation. For example in manufacturing, material purchases, production, and delivery processes can be categorised as core processes. The efficiency and uniqueness of these core processes gives a particular organisation a competitive advantage among other organisations. The support processes such as accounting, marketing, and human resource processes facilitate the smooth operation of the core processes. Support processes tend to be similar across organisations, irrespective of their core business activities. The management processes are the ones that plan and manage all the core and support processes. For example auditing, KPI (key performance indicators) reporting and planning are some of the management processes.
Another classification of business processes is based on the object that is associated with the process (Zur Muehlen, 2004c). In this view, processes are classified as *material*, *information*, and *business* processes (Medina-Mora, Wong, & Flores, 1993; Medina-Mora, Winograd, Flores, & Flores, 1992). A *material* process refers to assembling physical components and delivering physical products. An *informational* process denotes the activities that create, manipulate, manage, and provide information needed in an organisation. *Business* processes refers to market-centred descriptions of an organisation’s activities, implemented as information processes and/or material processes.

Rolland’s (1998) classification of business processes as *strategic*, *tactical* and *implementation*, is based on the organisational level in which processes take place. The *strategic* processes takes place in the highest level of the organisation, and are generally of an ad-hoc and creative nature. The *tactical* processes take place at the next level down and deal with planning the strategic business decisions. The lowest level processes are *implementation* processes, where planned processes are performed. These *implementation* processes are of a more prescriptive nature.

In *workflow* research (the definition of workflow and its association with business processes is defined in section 3.2.3 below) there is another classification of processes as *production*, *administrative*, *collaborative*, and *ad-hoc* (Alonso, Agrawal, El Abbadi, & Mohan, 1997; McCready, 1992). This classification comes from computer scientists, who are more interested in understanding the controlling factors (van der Aalst, 2002) associated with process automation. This classification combines some of the previously mentioned classifications. For example, differentiation as production and administrative processes is similar to Childe et al.’s (1994) core and support processes. Classification as ad-hoc processes is based on the characterisation of the way in which process actions are ordered or lack of prior ordering.

In the context of this research, the focus is mainly on business processes that are associated with informational objects and processes that are somewhat prescriptive in nature such as administrative or some production processes (Alonso et al., 1997). Some examples of such processes would be student enrolment in a tertiary institute or loan approval in a bank. In processes like this, it is important for certain tasks to take place in a particular order, with some leeway for flexibility.
Therefore, the key focus of this research is any process that is associated with an information object and has some prior understanding of the ordering of actions.

### 3.2.2 Workflow, WfMS, Process Definitions, and Instances

Process automation dates back to the 1970s when *office automation* was popular (zur Muehlen, 2004a). The meaning and usage of the term process automation has changed over the last three decades. In the era of office automation the simple use of e-mail, faxing documents, or computers for creating documents were considered examples of process automation. The idea was to use certain technologies to carry out certain activities of a business process. However, the process was still under the control of process participants who performed these activities and the routing decisions were made by humans based on the need (Georgakopoulos et al., 1995). For example, if the faxed documents were not clear due to some transmission errors, a human at the receiving end would request a re-transmission.

This ability for humans to control and coordinate process activities as the process exceptions take place or evolve had both advantages and disadvantages. The main advantage is the flexibility and ability in handling exceptions and deviations as suited by the process at hand. The disadvantage of humans coordinating process activities is the possibility of creating errors such as sending wrong documents or doing activities in illogical or wrong order.

The use of technology for the purpose of coordinating and controlling process tasks was introduced by workflow researchers (Caverlee, Bae, Wu, Liu, Pu, & Rouse, 2007). At present, the term process automation refers to creating a system that controls the business process activities according to a predefined set of rules. This leads to the definition of the term ‘workflow’.

According to the Workflow Management Coalition–WfMC (Feb 1999, p8), “*workflow is a automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules*. The WfMC consists of researchers and practitioners that have taken the initiative to set standards for workflow related
technologies. Hence, in the context of this research the terms that are defined by WfMC are considered as the industry accepted and widely used terminology.

Another term that is closely associated with a workflow is Workflow Management Systems or WfMS for short. According to WfMC (Feb 1999, P9) a WfMS is defined to be “a system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications”.

From mid 1980’s there have been many WfMS that were developed either as research prototypes or commercial applications (zur Muehlen, 2004a). Also the number of open source workflow management systems tends to increase by the day (Baeyens, 2004) as the demand and need for WfMS increases. The main reason for this interest in WfMS is its appropriateness for business process automation.

Organisations use WfMS to automate processes, mainly for easy management and control of tasks. The automation alone will reduce certain manual process steps such as record keeping, date stamping, logging, and tracking work (Plesums, 2002). In addition, Caverlee et al. (2007) lists the following to be other added advantages of automation:

- less chances of work getting misplaced or delayed,
- managers being able to focus on business issues rather than routine work assignment and control activities,
- work getting executed as planned by management, and
- parallel execution of activities that reduces the process time.

In the context of this research, the focus is on business processes that are automated using WfMS. These WfMS could be either commercial, open source or in-house web-based systems, used to automate business processes associated with an information object. The use of web technologies for process automation is further discussed in section 3.3.2.

The two terms process definition and process instances are closely associated with WfMS. Process definition is referred to as “the representation of a business
process in a form which supports automated manipulation, such as modelling, or enactment by a workflow management system” (WfMC, Feb 1999, p11). The processes are usually first defined using a visual process modelling tool and then converted to a machine-readable format for enactment.

A process instance refers to “a single enactment of a process, or activity within a process, including its associated data” (WfMC, Feb 1999, p15). For example, in a loan approval process, a loan application approval for a particular applicant is a process instance. Data that is associated with this specific loan application therefore would be associated with this process instance.

A large number of process definition languages have emerged over the last decade and attempts are made to standardise these process definition languages. Despite these standardisation efforts there are still a number of process definition languages. High-level visual process modelling tools and process definition languages are discussed in detail later in section 3.3.3.

3.2.3 Pragmatic, Semantic, Syntactic, and Implementation of Processes

Chapter-1 introduced a framework named paradigm of process automation-PoPA framework (Figure 1.1 in page 5), which identified various representations of business processes at different abstraction levels. There were four levels identified as pragmatic, semantic, syntactic, and implementation. For the completeness, an analysis of the usage of the key terminologies, in relation to this framework is presented below.

3.2.3.1 Contextual Artefacts in the Pragmatic Level

Pragmatic level represents contextual information elements that define the existence of a business process. In this research, therefore the term contextual information refers to various policies, laws, regulations, practices, guidelines, structures, Acts, and rules that define business process elements and their associations. These artefacts could be both internal and external to the organisation. This usage of the term context here is therefore similar to its usage by Ramesh, Jain, Nissen, and Xu (2005), where they propose a framework to manage process redesign based on contextual information evolution.
In other works, the term context is used differently. For example, Maus (2001) presents a dimension of workflow context space as shown in Figure 3.2. In comparison to Maus’s usage of the term, here context refers only to the causality and organisation spaces. Similar to Maus (2001) many other researchers also use the term workflow context to signify the environmental elements that are either organisational or technical (Judd & Steenkiste, 2003; Pohl & Haumer, 1997; Srinivasan, Ngu, & Gedeon, 1995).

![Figure 3.2- The Dimensions of the Workflow Context Space by Maus (2001)](image)

As mentioned before, the usage of the term context in this research denotes a broader and different concept, referring to both internal and external policies, procedures, structures, guidelines, Acts, regulations, and rules. In most organisations, these contextual elements are not in digital formats. Further, there are no standards used within and across organisations to create the contextual information elements (Salminen, Lyytikäinen, & Tiitinen, 2000). Irrespective of the disparity in presentation formats (digitised or hardcopy) and standards, these artefacts define the business processes (Ramesh et al., 2005). Thus, changes to these artefacts have a significant impact on the processes. For example, change in a policy that defines ‘leave approval’ would result in having to change the organisational leave approval process. If this leave approval process is automated, then changes to these high-level contextual artefacts (policy) needs to be reflected in automated systems.

Many research studies discuss the issue of managing the above-described contextual artefacts in a manner that those are useable in conjunction with
information systems or workflow systems. Kumar and Zhao (2002), Huan and Shan (1999), and Bussler and Jablonski (1995) discuss the use of organisational policies for managing participant allocation in workflow systems. In addition some explore mechanisms to structure organisational policies into definition languages for easy interpretation by information systems (Martin, Xie, & Yu, 2006; Kumar et al., 2002).

In relation to contextual artefact changes, many researchers attempt to find techniques and tool for managing change information systems. Tripathi and Hinkelmann (2007) introduce a methodology and system for managing changes in service-oriented-architecture based business processes. Srivastava (2007) presents a method called AutoSeek, that contributes towards a broad framework that could methodically make process improvements. However, as revealed by Ramzan and Ikram (2006) in their literature survey paper, there does not exist holistic solutions to the problem of managing contextual changes in implemented systems.

According to the usage of the term contextual artefacts, the organisational structures are also considered as context information that helps to shape a business process. In comparison to predominantly text-based contextual artefacts such as laws, regulations, Acts, and guidelines, organisational structures are easy to present in a computerised data structure. Hence, there are a number of studies (Russell et al., 2005b; Zur Muehlen, 2004b; van der Aalst, Kumar, & Verbeek, 2003a) that focus on modelling organisational structures in a manner that they are usable within WfMS.

In this research, the association of contextual artefacts with business processes or workflow systems is discussed in a different perspective. In here, the focus is on referencing to related contextual artefacts, so that a change triggers a change in the process models or the workflow elements. In other words, this research does not give an emphasis to formulating a wide range of contextual artefacts into any formal descriptions, as suggested by Ramesh et al.(2005) or Ali et al.(2005a), but only records a reference to these artefacts in its current form. It is worth noting that, in any case if a particular organisation is already using a sophisticated system that interprets contextual artefacts, this also can be supported by the solutions provided in this research.
3.2.3.2 Operational Business Process in the Semantic Level

Every organisation constitutes a number of business processes. These processes are not always visualised into models or automated in workflow systems (McCormac et al., 2005). However there is a common understanding among process participants on ‘how things are done’, by development and acceptance of mental models, or deeply ingrained assumptions, generalisations or pictures that provide an understanding of the way things are done (Senge, 1994). In the semantic level, of the PoPA framework represents the operational processes that are not necessarily visualised or automated.

Operational processes are organised in a manner that defines the functionally of that organisation. This aspect of signifying the organisational functionality was the main reason to use the term ‘semantic level’ to represent the operational business process. Also the levels of conceptual modelling as suggested by Boman, Bubenko Jr, Johannesson, and Wangler (1997) confirms this argument.

Issues in relation to processes identified at the semantic level are usually discussed in business or management research and not often under workflow or information systems research. Process re-engineering and process automation are two of the main research topic areas that discuss issues and provide solutions, to business management problems in process automation.

In the early eras of process re-engineering the focus was towards making radical changes to processes and revolutionising work practices (Hammer et al., 1993; Davenport, 1993; Hammer, 1990). As a result of these re-engineering efforts, it was advocated to use the advancements of information technologies to control and manage the processes (Davenport & James, 1990). However, later researchers (Kettinger et al., 1997) brought the concept of progressive redesigning that focused towards evolutionary modifications as opposed to the obliterate approach advocated earlier.

The evolution of business processes can be triggered by either internal factors or external factors. These factors include, new opportunities, external threats, increasing competition, or crises that refers to declining performances and cash flow (Caverlee et al., 2007).
In this research, the focus is towards understanding the impact of evolutionary changes in automated systems that support the changing business process. This is termed as Business Process Evolution Management (BPEM). A number of previous works discuss issues, challenges, and solutions in relation to BPEM.

Researchers such as Marjanovic (2007), Lu, Sadiq, and Governatori (2007), Lu, Sadiq, Governatori, and Brisbane (2006b), Rinderle, Weber, Reichert, and Wild (2005), Weber, Rinderle, Wild, and Reichert (2005), and Loossens and Van Houtte (1995) suggest the mechanism of using successful previous work practices, also known as previous cases, and associated data to carry out the evolutionary changes in business processes.

Ramesh et al. (2005) suggest the use of knowledge-based systems combined with a measurement-driven problem solving approach to diagnose process types (known as pathologies) and to recommend appropriate redesign transformations for the process. These knowledge repositories capture not only previous case histories, but also contextual information artefacts that define the processes.

Some researchers take the view that the evolutions of processes (and organisations) are driven by technological advancements. Based on this view, Aversano, Bodhuin, Canfora, and Tortorella (2006) suggest the concept of Joint Evolution of business Processes and software Systems (JEPS). JEPS integrates measurement, decision-making, and critiquing techniques for analysing business processes and identifying activities to be innovated using software systems.

The process represented in the semantic level denotes the perception of participants who perform the process actions. Hence, these operational processes are often enriched with tacit knowledge such as experience, mental models and cultures carried by the participants. Accordingly, participants collectively have the best knowledge repository and valuable insights on the process, which could lead to successful process innovations. With this viewpoint, Lewis, Young, Mathiassen, Rai, and Welke (2007) suggest using the perception of multiple stakeholders to guide the process innovations. The proposed approach captures, synthesises, and reconciles multiple stakeholders’ differing and potentially conflicting perceptions to yield a comprehensive foundation for business process innovation.
In the research presented here, the focus is towards suggesting process evolutions, based on changes that take place in contextual artefacts. The linking of contextual artefacts has similarities with Ramesh et al.’s (2005) work. However, this work also aims to find the propagating impact down to the implementation artefacts of the workflow systems, which is not covered in Ramesh et al.’s (2005) work. Further, this approach is more geared towards providing a holistic solution, which can handle process evolutions initiating at any of the four levels, pragmatic, semantic, syntactic and implementation; and in all four process elements, actions, participants and object data. In addition, the core objective of this research is not pointed towards suggesting any process innovations; rather to facilitate the carrying out of already identified changes in an efficient manner.

3.2.3.3 Process Models in the Syntactic Level

The syntactic level of the PoPA framework represents the models created to visualise the workings of the processes. In the practise of business process automation, modelling is an important phase. Modelling is an important phase in the practise of business process automation, and is carried out for comprehension and the streamlining of existing processes, then for automation purposes.

Capturing operational business processes into models is usually done using an agreed set of notations, which is suitable for the modelling task. Examples of such modelling notations are UML activity diagrams or EPC – Event-driven Process Chains (Details of modelling notations and languages are discussed later in this chapter). When using such notations, it is imperative to adhere to the syntax of the modelling notations. This gives the notion that representation of business processes using modelling languages is syntax-driven. In other words, the semantics of operational business processes can be captured within the expressive power of the modelling notations and its syntax. This aspect of syntactical representation of business processes is the rationale for naming the third level of the PoPA framework as the ‘syntactic level’.

These models created in process automation can change due to two reasons. Firstly, models may require changes due to propagating impact that flows down from pragmatic and semantic levels. Secondly, models may require to be changed due to design or modelling errors (van der Aalst et al., 2000).
When models are changed, assuring its correctness is researched under the topic workflow analysis. Workflow analysis refers to validation, verification, and performance analysis of processes (workflow analysis is discussed in section 3.4.2 of this chapter) (van der Aalst, 2002). This research focuses on identifying the components of the models that are changed prior to changes being carried out. In this sense, this research facilitates the process of workflow validation – which refers to assuring the created models are accurate from a business semantics point of view.

3.2.3.4 Web Workflow Artefacts in the Implementation Level

The implementation level of the PoPA framework refers to the automated business process. In this research, the focus is on business process automation using web-based workflow technologies.

In the automated world, processes are captured into numerous implementation artefacts. In web workflows, the types of artefacts include code files, data stores, templates, and various configuration files. Some research studies (De Silva & Ginige, 2007) in the web domain, categorise these elements and provide models to abstract the associations among these categories. In the author’s opinion, while these models are applicable to a certain extent, the association among web workflow artefacts is predominantly factored by the implementation approach.

Irrespective of the implementation approach, workflow artefacts are required to reflect the changes that take places in upper levels (pragmatic, semantic, and syntactic) of the PoPA framework. When changes are introduced to one implementation artefact, it would create ripple effects, due to its associations with other artefacts.

This section described a number of key terminologies and the usage of those terms within this research. The next section discusses the theoretical framework of this research.

3.3 Theoretical Framework

This research is closely related to Business Process Automation (BPA) and Business Process Evolution Management (BPEM). The term BPA refers to initial automation of organisational business processes using workflow technology. BPEM
denotes the subsequent modifications of already automated workflow systems, in order to reflect the changes that occur in operational business processes.

The core focus of this work is on impact analysis of process evolutions, which is identified as one of the steps in BPEM (see Figure 3.3 – created based on ideas drawn from works of Kettinger et al. (1997) and Georgakopoulos et al. (1995)).

Both BPA and BPEM involve a number of interrelated steps as indicated in the broader theoretical framework. Among these steps, evolution impact analysis is carried out immediately followed by evolution diagnosis and prior to changing models or workflow systems (details of these steps are discussed later).

Figure 3.3- Positioning of this Research in BPA and BPEM. Adapted from Kettinger et al. (1997) and Georgakopoulos et al. (1995)

Most steps shown in Figure 3.3 have been researched over many years; therefore can have more than one Methodology, Technique, and Tool (MTT) associated with each step. The placing of the theoretical framework in a cloud of MTTs in Figure 3.3 indicates this aspect of having numerous MTTs. Prior to investigating steps in BPA and BPEM, the general usage and meaning of the terms methodologies, techniques, and tools are presented below.

The term methodologies generally refers to the highest-level of abstraction for conceptualising problem solving approach (Kettinger et al., 1997). For example,
in software application implementation, the development team may decide to either use structured or object-oriented methodologies for analysis, design and development (Fraser, Beck, Booch, Constantine, Henderson-Sellers, McConnell, Wirfs-Brock, & Yourdon, 2005). As cited by Kettinger et al. (1997) a more appropriate definition for the term methodology is a collection of a set of principles and a common philosophy for solving a targeted problem, such as process automation (Checkland, 1981). For instance, in process automation, there are two methodologies used to develop workflow systems; first is the method of embedding the process, in application code and second is the approach of abstracting the process, into a machine-readable process definition and enacting it with the support of a workflow management system (van der Aalst, Hee, & Hee, 2002a). Similarly, for each step identified in the above Figure 3.3 may have at least one methodology that defines the way in which the activities in that step could be achieved.

A technique is the next level of details in abstraction of a problem solving approach (Kettinger et al., 1997; Hackathorn & Karimi, 1988). According to Hackathorn and Karimi (1988) the term technique is a procedure for accomplishing a desired outcome, which specifies the steps, as well as the necessary inputs and results of each step. For example, in process automation that deploys the model driven architecture (MDA) methodology, the process models will be first created and these will be used for generating the process-oriented applications (Brambilla, Ceri, & Fraternali, 2006). In this example, it explains the technique that has two steps involved in using the MDA methodology.

A tool is referred to be particular instrument that may be in the form of a software, document template, modelling notations or a simple set of guidelines that could be deployed for the purpose of carrying out the task steps identified in a related technique (Kettinger et al., 1997; Palvia & Nosek, 1993; Hackathorn et al., 1988). For example in process automation, process modelling can be done using a number of modelling tools such as UML activity diagrams (OMG, August 2005), Petri-Nets (Working-Group, 1997) and Event driven Process Chains (EPC) (Mendling, Neumann, & Nuttgens, 2005a).

Next, the steps of BPA and BPEM (in Figure 3.3) are discussed. In this discussion, some associated MTTs are examined in detail. However, some of the
MTTs that are closely related to this research will be presented separately in section 3.4. Further, as some steps are self-explanatory and loosely related to this research, the discussion on those is kept to a minimum.

3.3.1 BPA and BPEM

This section discusses each step associated with BPA and BPEM (Figure 3.3) in the logical order, denoted using arrows in the figure.

3.3.1.1 BPA Step – Process Identification

First step in BPA is identifying or selecting certain processes in an organisation for automation (Kettinger et al., 1997). While there are not many scientific MTTs for identification of processes for automation, it can be driven by many business factors such as new opportunities, external threats, increasing competition, or crises that refer to declining performances and cash flow (Caverlee et al., 2007). Usually processes with inefficiencies are selected for the automation, with the objective of improving the performance of them. Some argue that automation alone facilitates in reducing certain manual process steps such as record keeping, date stamping, logging, and tracking work (Plesums, 2002).

3.3.1.2 BPA Step – Process Re-Design (Re-Engineering)

Once the processes are earmarked for automation, these may go through a step of redesign. Business Process Redesign (sometimes referred to as re-engineering) - BPR has been popular since the 1990s. BPR is carried out for a number of reasons, such as: to increase customer satisfaction, improve efficiency of business operations, increase quality of products, reduce cost, and meet new business challenges and opportunities by changing existing services or introducing new ones (Georgakopoulos et al., 1995).

BPR initially focused on radically changing business processes to gain maximum advantage (Hammer et al., 1993; Hammer, 1990). Later in BPR the emphasis was towards introducing incremental changes into a business process at a pace that is appropriate for the organisation and to still gain effective results (Jarvenpaa, Stoddard, & Rhetoric, 1998; Stoddard et al., 1995). In Figure 3.3, the step BPR refers to reorganising the process elements, but not necessarily automating the process (Georgakopoulos et al., 1995).
3.3.1.3 BPA Step – Create Process Models

This step refers to abstracting the operational business processes into conceptual models. This conceptualisation helps to refine the processes further, as models facilitate visualising the inner workings of the process.

Generally, there are two types of models created in process automation. Initially visual models are created using visual modelling notations that is supported by the selected WfMS. Secondly, these visual models are converted into machine-readable models, which are enacted by the WfMS.

There are a number of modelling tools that can be used for the purpose of modelling both high level and machine-readable process models. The selection of the appropriate modelling tool, largely depends on the WfMS selected for process automation (Smith & Fingar, 2003). However, it is worth noting that there have been some efforts by the WfMC and other institutes to standardise workflow modelling languages. Further details in relation to process modelling tools and its standardisation efforts are later discussed in section 3.3.2.

3.3.1.4 BPA Step – Analyse Process Models (Workflow Analysis)

After process models are created, prior to implementation, these models are analysed in order to verify the correctness of process models. Some refer to this step of process model analysis as workflow analysis (van der Aalst, 2002).

Workflow analysis constitutes three analytical concepts, which are validation, verification, and performance analysis (van der Aalst, 2002). Validation refers to assuring the modelled process is exactly what is required by the business. Usually validation is supported by process simulation (van Hee, Oanea, Post, Somers, & an der Werf, 2006; Bosilj-Vuksic, Jaklic, & Popovic, 2005; Padmos, Hubbard, Duczmal, & Saidi, 1999) facilities provided by the WfMS. Verification refers to ensuring the structural and syntactic correctness of process models according to the process modelling notations or language used. Different researchers (Sivaraman & Kamath, 2005; Eshuis & Wieringa, 2002; Marjanovic, 2000; van der Aalst, 1997) use various approaches and algorithms for verification purposes. Finally, the performance analysis refers to understanding any bottlenecks or any other performance issues. Similar to verification, there are a number of approaches and mechanisms (Stefanov & List, 2005; Aiello, 2004) used for performance
analysis. Workflow analysis would lead to refinement of process models or further redesign of the process in order to rectify any identified errors or inconsistencies.

An extensive study of previous works in the area of workflow analysis is presented in section 3.4.

3.3.1.5 BPA Step – Implement Process Models

Process model implementation refers to setting up data prior to end-user testing of automated processes. This includes tasks such as populating any reference data tables and/or integrating with other systems that need to interact with the WfMS. These implementation tasks are highly dependent on the WfMS used for automation purposes.

The above-discussed steps (3.3.1.1 to 3.3.1.5) identify the typical procedural steps involved in the initial automation of a process. The next set of steps shows the tasks involved in subsequent modification of already automated processes, which is referred to as BPEM.

3.3.1.6 BPEM Step – Evolution Diagnosis

The phase of BPEM begins by identifying the changes required in operational business processes. These changes can be driven by strategic business plans, regulatory needs, operational reasons, design and logical errors that occurred in the initial modelling and technical issues with the automated system itself (van der Aalst et al., 2000). This diagnosis step denotes the identification and acknowledgement of the required changes by process managers and business analysts (Kettinger et al., 1997).

3.3.1.7 BPEM Step – Impact Analysis of Business Process Evolution

Once requirement for changes are identified, planning the task reflecting these changes is denoted by the impact analysis step. In other words, evolution impact analysis is the task of assessing the extent of change, by identifying likely ripple effect of evolution. The expected output of this step is a complete plan that lists the artefacts that require changes (Yau, Collofello, & MacGregor, 1978). These artefacts could include parts of process model/s and implementation artefacts.
In practice, the boundary or scope of the first two steps in BPEM – evolution diagnosis (discussed in the previous step) and impact analysis, would be somewhat blurred. The boundary between these two steps depends on who is more actively involved in it. For instance, the previous step of evolution diagnosis would be of much interest to process managers and organisational decision makers, who release resources for workflow system changes. The step of evolution analysis is of interest to the business analysts who are responsible for successfully reflecting required changes in workflow systems.

Despite this step being perceived, as unnecessary in relatively small processes, there are a number of advantages in having a unique step for evolution impact analysis. Planning required changes is vital when processes are relatively critical, large and complex (Sadiq, Orłowska, Sadiq, & Foulger, 2004). Further, in reality changes are not always carried out by business analysts who were involved in the initial process automation. This means business analysts’ lack of understanding of the inner workings of the process could lead to costly errors and inconsistencies.

The term evolution impact analysis (sometimes referred to as decision and risk analysis (Beydeda & Gruhn, 2001)) is different to the term workflow analysis (van der Aalst, 2002). Workflow analysis, which is discussed under BPA, focuses on validation, verification, and performance analysis of process models. In contrast, evolution impact analysis denotes the assessment of risks and impacts of changes prior to them being introduced to process models (Bodhuin et al., 2004).

The steps of business process evolution impact analysis and workflow validation have the same end goal. This end goal is to assure that the modelled process is what is required by the organisation. However, the major difference between these steps is in approaches and techniques being used. Workflow validation mainly uses simulation for assuring the consistency of the process. However, simulation greatly relies on the human business analyst’s ability to recognise any differences between the required and modelled process. In addition, the simulation is only as good as the data set that is used for the simulation purposes (Jansen-Vullers & Netjes, 2006). In contrast, methodologies used in an evolution impact analysis, identify the risks of changes, with the objective of minimising subsequent errors and inconsistencies. When proper tools are used for this impact analysis task, it can be assured that the modelled process is free of human errors.
Change impact analysis is highly researched and practiced in the software engineering domain (Bodhuin et al., 2004; Zhao, Yang, Xiang, & Xu, 2002). For example, the mini-cycle of change as described by Zhao et al. (2002) and Yau et al. (1978) indicates the need for a planning phase consisting of program comprehension and change impact analysis. Similarly, process impact analysis gives the opportunity for business analysts to comprehend the already automated process and find the propagating impact.

This aspect of process evolution analysis is a largely neglected area in workflow research (Bodhuin et al., 2004). This gap has been identified and recently some work has been proposed in this area (Ramesh et al., 2005; Soffer, 2005; Bodhuin et al., 2004). While these studies attempt to address some issues of impact analysis, still problems exist in this domain. In particular, most previous research fails to provide a holistic solution, which can manage evolution of process actions, participants, and business objects. In addition, the ability to manage any type of change, which could initiate at the pragmatic, semantic, and syntactic levels, down at the implementation level. Further, in previous research (even in workflow evolution) the solution is around a particular technology. This research therefore focuses on finding a holistic solution to the problem of impact analysis of business process changes, which would facilitate reflecting process changes rapidly and effectively as required by the organisation.

Further exploration and in-depth studies into more previous work in the areas of workflow analysis and impact analysis are presented later in this chapter.

### 3.3.1.8 BPEM Step – Modify Process Models

Modification of process models is the next logical step that follows the previously discussed evolution impact analysis. There are two other alternative steps (these are discussed separately), that could follow impact analysis. First is the possibility of impact analysis leading into a process redesign (Ramesh et al., 2005) as indicated in Figure 3.3. Similarly, if the changes were identified to be technical changes, those would be reflected directly in the implemented workflow artefacts, but not in models. This section however discusses MTTs of process model alteration.

The changing of already automated process models is highly researched under the terms such as workflow evolution, adaptation, flexibility, and dynamism.
These terms are sometimes overloaded with two ideas that relate to process evolution. Researchers identify two aspects to evolution as static and dynamic evolution (Caverlee et al., 2007; Casati et al., 1998). The static evolution denotes changing process definitions to reflect the business process changes and dynamic evolution refers to managing running instances of a workflow whose definition being changed (Casati, 1998). The step of changing process models exclusively represents the concept of static evolution.

The success of both static and dynamic evolution depends on the flexibility of the process modelling tools and WfMS. Even after many years of research some identify the inflexibility of WfMS as one of the key issues in workflow research (Caverlee et al., 2007; Zur Muehlen, 2004b, 2004c; Aiello, 2004; Adams, Edmond, & ter Hofstede, 2003; van der Aalst, 2002; Nutt, 1996; Veijalainen, Lehtola, & Pihlajamaa, 1995). Encouraged by this lack of support for adaptability in WfMS, many continue to explore this area in search of possible solutions.

Researchers such as van der Aalst and Jablonski (2000), Casati et al. (1998), and Nutt (1996) provide insights into fundamental theoretical concepts in relation to workflow evolution. There are mainly three types of static evolutions identified. These are extensions that refer to additions of elements into process models, reductions denote the deletion of elements from process models, and re-linking identifies the modifications that could include both deductions and extensions (van der Aalst et al., 2000).

Though workflow adaptability is vital, from this research’s point of view it is not the core focus of this thesis. Nevertheless, the author expects particular WfMS to be reasonably adaptable such that it does not create runtime errors, when changes are reflected.

3.3.1.9 BPEM Step – Commit Workflow Changes and Manage Running Instances

Closely related to static evolution (changing of process models) is the issue of managing running instances within the WfMS, also referred to as dynamic evolution (Casati et al., 1998). The interest on studying dynamic evolutionary aspect of workflows has been on increase since the late 1990s. For instance, researchers
such as Rinderle et al. (2006; 2005; 2004; 2003), Weske (2001b), and Reichert and Dadam (1997) have provided solid groundings for dynamic evolution of workflows.

The issues in relation to dynamic workflow evolution are greatly important and applicable in the wider theoretical framework presented in Figure 3.3. However, its applicability to this research is limited. As such, the support for dynamic evolution is not discussed in detail in this work.

3.3.1.10 BPA and BPEM Step – Evaluation and Testing and Rollout

The last two steps common to both BPA and BPEM (see Figure 3.3) are evaluations and testing, and rollout. These steps are self-explanatory for readers who have been involved in any scale software development efforts. These steps are added to this theoretical framework purely for completeness. While there can be a number of MTTs applicable to these steps, the author does not attempt to summarise those work in this research, as these are clearly beyond the scope of this work.

This section discussed the steps presented in the theoretical framework (Figure 3.3) that this research is founded on. In this discussion, some landmark research works are briefly mentioned depending on their applicability to this research. The following two sections discuss workflow technologies and process modelling concepts in detail with an in-depth review of relevant literature.

3.3.2 Workflow Technology

For the automation of business processes WfMS are currently considered to be the preferred method. The steps involved in using WfMS for BPA are highlighted in the theoretical framework presented in Figure 3.3. Irrespective of these steps, WfMS could be implemented using a number of different technologies. This section explores these different technological concepts associated with WfMS.

Similar to any software system, WfMS can be categorised in a number of ways. Generally, this categorisation is based on either, implementation architectures, features and functionality, code distribution method, or technology.

As mentioned previously WfMS is defined to be a system that enables defining and managing the execution processes (WfMC, Feb 1999). Based on this definition, such system can be categorised into four types of WfMS as: pure WfMS,
WfMS embedded in other systems, custom-made WfMS, and hard-coded WFM solutions (van der Aalst, 2005).

Pure WfMS are dedicated only for the purpose of workflow related activities such as creating process definitions and enactment. These systems might also provide advance facilities for process analysis and mining based on the logs and simulation.

Enterprise wide information management systems such as ERP systems also have certain workflow features embedded into them. For example, SAP WebFlow is the workflow component of SAP offering all the functionality typically present in traditional stand-alone WfMS (van der Aalst, 2005).

In addition, many organisations have chosen not to use the above-mentioned commercially available WfMS, but build an organisation-specific solution. The reasons for this custom-made WfMS could be manifold such as high-price tags, restrictive and prescriptive nature of commercial WfMS, incompatibility with existing legacy systems and for research and analytical purposes. Some researchers argue that these custom made solutions only support a subset of the functionality offered by the first two categories (van der Aalst, 2005).

The other category of workflow systems are process-oriented applications where the processes are hard-coded in the applications to support a specific process (van der Aalst, 2005; van der Aalst et al., 2002a). These applications do not provide a generic framework for more processes to be added or existing processes to be changed, unless by modifying the underlying application code.

All four types of the above-mentioned WfMS (pure WfMS, WfMS embedded with other systems, custom made WfMS and hard-coded process-oriented applications) come in two flavours – commercially supported and open source systems (Baeyens, 2004). Some examples from many hundreds of open source WfMS are Open Business Engine2 a Java workflow engine, Twister3 a Java based WfMS that supports web services and WfMOpen4 a J2EE based implementation. Examples of commercial WfMS are Staffware Process Suite, FileNET BPM Suite, i-
Flow, FLOWer, WebSphere MQ Workflow (formerly known as MQSeries Workflow) and TIBCO InConcert (van der Aalst, 2005).

One of the notable advancements in WfMS is the use of web-technologies for its implementation purposes. As this research is focused on web-based WfMS, details of these systems are presented in the following section.

### 3.3.2.1 Web-based WfMS

With the development of web and related technologies, WfMS are also being ‘webified’ in order to gain cost effective and pervasive advantages that the web offers. Most commercial WfMS have recently added features that allow process to be extended as web-based workflow systems.

In the domain of Web Engineering (a term coined by Ginige and Murugesan (2001)), researchers have been exploring advanced topics that make it easier for processes to be automated using web-based technologies. For example, Barna, Frasincar, and Houben (2006) provide a mechanism for creating web application elements based on workflow models and Brambilla et al. (2006) present a mechanism for model driven web application development. However, these works mainly considers the process logic to be embedded with web application logic, thus fall short of providing the other advantageous features that pure-WfMS (van der Aalst, 2005) offer, such as easy modification of process definitions, exceptions handling, and dynamic changing of cases.

Accurately identifying the advantages of having features of pure-WfMS developed using web technologies, projects have emerged that focus on combining the two. Some of these ‘webified’ pure-WfMS are: WebWork\(^5\) (Miller, Palaniswami, Sheth, Kochut, & Singh, 1998), WWWWorkflow (Ames, Burleigh, & Mitchell, 1997), ADOME-WFMS (Chiu, Li, & Karlapalem, 2001c), Collaboration Management Infrastructure – CMI (Ngu, Georgakopoulou, Baker, Cichocki, Desmarais, Bates, & Technol, 2002), and LBPF (Liang, Marmaridis, & Ginige, 2007).

One of the early web-based WfMS is WebWork, which combines advanced features of WfMS such as build-time and run-time environments, code generation based on visual process models, role based access control and security via encryption of documents (Miller et al., 1998; Miller, Fan, Sheth, & Kochut, 1997). WebWorks

supports a distributed implementation with participation of multiple Web servers (see Figure 3.4).

Figure 3.4- The Distributed METEOR2 Model of WebWorks (Miller et al., 1998)

WWWorkFlow is another web-based WfMS that was developed in 1997 (Ames et al.), and is the same as WebWorks. The separation of process mediation from product data management is advocated as a special feature of the WWWorkFlow architecture (see Figure 3.5).

Figure 3.5- WWWorkFlow System Architecture - adopted from Ames et al. (1997)
Another notable contribution for web-based WfMS is ADOME-WFMS, by Chiu, Li, and Karlapalem (2001b; 2001c; 2001a), which deploys object-oriented concepts for modelling and implementation. The architecture and functional aspects of ADOME-WFMS is as shown in Figure 3.6.

![Figure 3.6- ADOME-WFMS architecture - adopted from Chiu et al. (2001c)](image)

With the interest and focus on managing dynamic evolution of web workflow systems Liang, Marmaridis, and Ginige (2007) provide a lightweight web-based WfMS that binds the process definitions with each workflow instances. The core contribution of this Lightweight Business Process Flow (LBPF) engine is the facility for handling multiple versions of the same process definition. In addition, process definitions can be modelled using multiple methods such as graphical notation or text or tabular based MS Excel files (see Figure 3.7).
Figure 3.7- Logical Architecture of LBPF engine - adopted from Liang et al. (2007)

Similar to the above presented web-based WfMS, there are a number of other systems, which focus on different aspects. For instance, the WASA (Workflow-based Architecture to support Scientific Applications) focuses on creating a dynamic environment, which supports issues which are unique to scientific workflows (Vossen, Weske, & Wittkowski, 1996). Ahn et al.’s (2000) work concentrates on providing security to existing web-based WfMS, using Role Based Access Control (RBAC) mechanisms. There are many other web-based WfMS, which are somewhat similar to the above presented set. Thus, these are not covered in detail in this work.

The implementation mechanism of functional components of web-based WfMS and operational business processes has a direct correlation. Understanding this correlation is important for reflecting changes to processes in web-based WfMS. The association among implementation artefacts of a web-based WfMS largely depends on the system architecture. This association among web system artefacts creates a ripple effect when physically changing the implementation artefacts.

Providing a general abstraction of the associations among component parts of web-based WfMS is sparsely researched. However, this notion of correlation among web-workflow artefacts needs to be taken into consideration, when providing a holistic solution for management of business process evolutions in web-based WfMS. One of the notable contributes comes from De Silva et al. (2007). The components, referred to as aspects of a web application are identified to comprise of five types of models as: Hypertext model, User model, Process model, Data model
and Presentation model (De Silva et al., 2007). The interrelation between these artefacts are clearly presented by De Silva et al. (2007), as shown in Figure 3.8 below.

These aspects (or component parts) of web-based WfMS, could be implemented using a multitude of web-based technologies. For example, presentation and hypertext models are generally implemented using a combination of HTML, Java applets and style managing methods such as CSS (Miller et al., 1998). Workflow models generally use XML based process modelling languages, such as XPDL – XML Process Definition Language (details on process modelling methods are discussed in the next section). The task model of the system may use CGI or Java applications (Miller et al., 1998). Further, there is the possibility of tasks to be implemented as functions of existing external legacy information systems. The data model can be implemented using varying technologies with different granularity such as printable document formats like Microsoft Word, Excel, and PDF or structured formats similar to XML documents or database tables or objects or a combination of above.

Web services and associated technologies have recently become popular for web-based workflow implementations (Liu, Ngu, & Zeng, 2004; Zeng, Benatallah, Lei, Ngu, Flaxer, & Chang, 2003; Zeng, Benatallah, & Ngu, 2001). According to the working group specification issued by W3C-WWW Consortium (Booth, Haas, McCabe, Newcomer, Champion, Ferris, & Orchard, 2004) a web service provides the capability for different software applications, running on a variety of platforms and/or frameworks to interoperate. In process automation the above-mentioned
interoperability gives the added advantage for inter organisational collaboration and to integrate heterogeneous legacy systems in process automation.

The terms choreography and orchestration are closely associated with web services and describe two aspects of modelling processes from composite web-services. Orchestration refers to an executable business process that can interact with both internal and external web services; and choreography defines a more collaborative approach that allows each involved party to describe its part in the interaction (Peltz, 2003). With the increased use of web services for process automation, Business Process Execution Language for Web Services, commonly referred as BPEL (Andrews et al., 2003), has become popular for orchestration of processes for automation purposes (details of BPEL and other process modelling languages are discussed in detail in the next section).

There are many differences between WfMS from the implementation point of view (Smith et al., 2003; Maus, 2001). In the interest of standardising workflow technology, WfMC has proposed a workflow reference model that conceptualises the component parts of a WfMS and their interaction with other systems (Figure 3.9).

![Figure 3.9- Workflow Reference Model (WfMC, 1999)](image)

In addition to the workflow reference model, there are a number of other concepts that WfMC is attempting to standardise. There are a workflow meta-model and a process definition language. Details of these other WfMC related technological concepts are discussed in subsequent sections.
In this research, the focus is on web-based WfMS that uses any technology combination for implementation purposes. There are two types of associations that need to be identified; the association of process elements (actions, participants, and object data) with relevant web-based workflow artefact (such as CGI applications, HTML forms and database or documents) and the association of the artefacts among themselves. The requirement is however to comprehend the associations between technological concepts, in order to accurately find the propagating impact of a change.

One of the core aspects of using WfMS for process automation is the conceptualisation of processes using appropriate modelling techniques. This conceptualisation enables an understanding of the nature of associations among process elements. The MTTs used for business process modelling is explored in the next section.

### 3.3.3 Business Process Conceptualisation

Conceptualisation of processes is generally referred to as process modelling. In this sense, the interest in process modelling dates back to 1917, when Henry L. Gantt created the popular project management tool called Gantt Chart (Whitten et al., 2007b). A Gantt Chart is the most simplest and earliest version of a process model that shows the interrelationships between tasks (Breton & Bézivin, 2000) performed to achieve a particular goal. Associated with the Gantt Chart are the Project Evaluation and Review Technique, also commonly referred to as PERT charts, which was developed by the U.S. Navy for controlling large weapon development projects in the 1950s (Whitten et al., 2007b). These tools mostly focus on showing the association of activities with time points. In addition, these models only provide the two simplest flow control patterns – sequential and parallel.

Today’s complex business processes have a number of intricate concepts that cannot be visualised using simple Gantt or PERT charts. For instance, illustrating complex flow control patterns (Russell, ter Hofstede, van der Aalst, & Mulyar, 2006a; van der Aalst, ter Hofstede, Kiepuszewski, & Barros, 2003b) and action association with participants cannot be straightforwardly reflected in the above-mentioned tools (Breton et al., 2000). For this reason, researchers and practitioners...
have been endeavouring to discover suitable methods for process conceptualisation purposes.

There are three aspects to conceptualisation of processes. These are meta-models, visual high-level models, and machine-readable models. These three concepts are interrelated. According to Atzeni, Cappellari, and Bernstein (2005) a meta-model is a set of constructs that can be used to define models, which are instances of the meta-model. In this context, there are meta-models associated with both visual and machine-readable models. Then in a single process automation exercise, it might be required to use a combination of both visual high-level and machine-readable tools (Hollingsworth, 2004; van der Aalst et al., 2003c; Weske, 2000). First the conceptual models of the process required to be created using visual notations and then machine-readable process models would be created for enactment by the WfMS (Hollingsworth, 2004; Weske, 1999). The next section discusses these three concepts separately, with the objective of comprehending all possible associations among process elements.

3.3.3.1 Meta-Models for Business Process Abstraction

One of the important concepts associated with most process modelling is meta-models (Breton et al., 2000). There are a number of meta-models proposed for the purpose of conceptualisation of business processes. The evolution of some of these process meta-models are presented in chronological order.

One of the first meta-models is Workflow Activity Model- WAMO (Eder & Liebhart, 1995), which conceptualises the types of associations among process activities (Figure 3.10). This model captures action, data, and participant aspects of processes into activity, form, and agent components respectively.
Chapter 3 - Literature Review

The focus of the Dynamic Workflow Management – DWM meta-model (see Figure 3.11) is to capture the behavioural aspects of the business process (Kwan & Balasubramanian, 1997). Apart from the behavioural aspect, this meta-model also captures the functional (process actions), informational (object data) and organisational (participant) perspectives in relation to a process.
process instances respectively. In each of these levels, the association among process actions, participants (problem solving agents – PSA roles), and Event Condition and Action (ECA) rules are shown in Figure 3.12. Another special feature of this ADOME-WFMS meta-model is capturing capabilities of PSA roles, which is not adequately covered in other models.

In an attempt to standardise workflow technology, the WfMC published a meta-model in 1999 (see Figure 3.13). While WfMC’s model does not introduce any new aspects to the workflow meta-model, it cohesively combines several ideas from previous meta-models. The WfMC’s meta-model presents the possibility of process actions and workflow production data to be represented in external systems. WfMC’s meta-model has been used as a foundation for many workflow related research work. For instance, Lehmann (2006) extends the WfMC’s meta-model with data access capabilities. Further some (McGregor, 2002) have also added extensions to this meta-model for business process performance analysis.
This aspect of extension of workflow meta-models for various analytical purposes is the newest focus of research in the meta-model area. For example, Stefanov and List (2005) extends Event-driven Process Chain (EPC) meta-model for performance evaluation purposes (see Figure 3.14).
The other form of extensions of workflow meta-models concentrate on handling exceptions and cases or process instances. This approach of introducing exceptions and instances were initially introduced by Chiu et al. (1999) and Breton and Bézivin (2002). However the recent model proposed in van der Aalst et al. (2005d), provides a much refined version of this meta-model (see Figure 3.15).

Some researchers believe the success of a process definition language begins with a solid foundation laid by a proper meta-model that captures all possible behavioural aspects of a process (Breton et al., 2000).

For sometime WFMC’s meta-model was considered the standard. However, recently some researchers have argued the limitations of the WFMC’s process definition meta-model. As a result there has been attempts to either provide a suitable meta-model (Breton et al., 2002; Breton & Bezivin, 2001; Breton et al., 2000) or to extend the WFMC’s process definition meta-model (Lehmann, 2006). Breton and Bezivin (2000) argue that WFMC’s process definition meta-model is limited in capturing the behavioural and data aspects of a process. Then they propose a meta-model that binds the WFMC’s process definition meta-model with the OMG Workflow Management Facility (Breton et al., 2002). Pointing to the lack of support...
for the data aspect in WfMC’s process definition meta-model, Lehmann (2006) proposes an extension to it.

Workflow meta-models give an insight into all possible correlations among process elements, which is vital in finding the propagating impact of process element evolutions. In this view, the knowledge drawn from the aforementioned process models is vital in formulating a solution in finding impact of process changes, which will be covered in later chapters.

3.3.3.2 Visual Process Modelling Tools

Visual models are significant, in relation to process conceptualisation. Visual models allow business analysts to comprehend and communicate internal workings of the process. In addition, visual models also facilitate process managers to understand any issues in relation to their operational processes. In process automation, visual models are also used for creating machine-readable processes, which are used by workflow engines for enactment. This section presents widely used, prominent visual process modelling tools.

There are a number of visual tools advocated for conceptualisation of business processes. From this plethora of visual modelling tools, six are selected for detailed discussion. These are Workflow-Nets (WF-Nets), a Petri-Net based tool, Event-driven Process Chains (EPC), Workflow-Graphs (WF-graphs), Unified Modelling Language’s (UML) Activity Diagrams, and Business Process Modelling Notations (BPMN). These six are selected due their wide usage in research work.

Petri-Net is a graphical and mathematical modelling tool, which consists of places, transitions, and arcs that connect them (Reisig, 1985). Petri-Nets were introduced by Carl Adam Petri's dissertation, in the area of automata in 1962. Since then there has been many extensions to Petri-Nets. Use of Petri-Nets for workflow modelling is an area of interest to this research. Business process conceptualisation using Petri-Net based model is referred to as workflow-nets (WF-nets) and exemplified in Figure 3.16.
WF-Nets are the most commonly used Petri-Nets based process modelling notation used for business process conceptualisation. WF-nets are extended with concepts such as ‘time’, which is not present in original Petri-Nets. The use of WF-nets is advocated for its sound mathematical support that facilitates complex analytical tasks (van der Aalst, 2002, 1997). However, only the original Petri-Nets are backed up with solid analytical algorithms. Therefore, the new extension of Petri-Nets, such as WF-nets, needs to be converted to original format for analytical purposes.

Lack of support for the data aspect is considered to be one of main disadvantages of WF-nets (van der Aalst et al., 2003c; Ruiz, van Harmelen, Aben, & van de Plasche, 1994). In addition, conceptualisation of processes using WF-nets is not a trivial task, unless the modeller is thorough with the notations and their usage. For that reason, WF-nets are not commonly used by business people for modelling of business processes.

Event-driven Process Chains (EPC) were originally introduced in 1992 by Keller, Nüttgens and Scheer within the framework of ARIS to model business processes (cited in Nüttgens, Feld, & Zimmermann, 1998). EPCs consist of richer semantics that allows capturing events, functions, organisational units that are
capable of performing the functions and data association with functions (see Figure 3.17).

![Figure 3.17- An Example of Process Modelling using EPCs (Nüttgens et al., 1998)](image)

While EPC has rich tools that are capable of capturing semantics of a business process, it lacks formalisms that allow it to be used for analytical purposes. For this reason, EPC models are transformed into Petri-Net based WF-nets for analysis (van der Aalst, 1999). However, in this transformation certain information such as data associations is dropped, as Petri-Nets do not exclusively support these concepts (Oren et al., 2005). In addition, there are certain extensions proposed (Mendling et al., 2005a) for EPCs to support a number of workflow patterns (van der Aalst et al., 2003b), which are not generally supported by original EPCs. Despite, the lack of formalisms, EPCs is a widely used tool for process conceptualisation purposes, due to its rich semantics.

UML activity diagram is one type of diagrams in a suite of modelling notations created by the Object Management Group\(^6\) (OMG). Due to strong partnership of commercial vendors in OMG, UML activity diagram has become one of the widely used business process modelling notations. Similar to EPCs, UML

activity diagrams also provide richer semantics for conceptualisation of processes (see Figure 3.18).

**Figure 3.18- UML Activity Diagram with Activity Partitioning to Swim Lanes (OMG, 2007)**

UML activity diagrams can model concepts of process actions, participants, and data. There is some debate over UML activity diagrams ability to capture advanced flow control patterns and provide formal verifications for process models (Dumas & ter Hofstede, 2001). There are many research (Eshuis & Wieringa, 2004; Eshuis et al., 2002; Eshuis, 2002) that provide mechanisms for the formal verification of UML activity diagrams. Also there are some work (Korherr & List, 2006; Nüttgens et al., 1998) that compares and suggests transformation between EPC and UML activity models.

Another popular process modelling tool is WF-graphs, which is a form of directed acyclic graphs (DAG) (Sadiq & Orlowska, 1997). In WF-Graphs vertices represent nodes and directed edges represent transitions (see Figure 3.19). WF-Graphs claims to confirm to WfMC’s reference model (WfMC, 1999) for workflow modelling (Mangan & Sadiq, 2002a). Further works based on WF-graphs concentrate on; workflow transformations (Sadiq & Orlowska, 2000c), verifications of structural correctness (Sadiq & Orlowska, 2000b), verification of temporal constraints of process models (Marjanovic, 2000) and validation of process constraints for flexibility (Sadiq, Orlowska, & Sadiq, 2005).
Figure 3.19- WF-Graphs for Process Modelling (Sadiq et al., 2000c)

Business Process Modelling Notations (BPMN)\(^7\) is a visual modelling notation standardisation effort initiated by Business Process Management Initiative (BPMI). With the recent merger between Object Management Group (OMG) and BPMI, the notations are now maintained by the OMG. BPMN notations have several levels of modelling notations and also facilitate inter and intra organisational process modelling (OMG, 2004). UML modelling notations (in particular UML activity diagrams) have a strong influence on BPMN notations (see Figure 3.20).

Figure 3.20- BPMN Process Example (OMG, 2004)

Similar to meta-models, high-level visual models also gives insight to various types of associations that could exist among process elements. Further, the associative aspect among process actions such as sequencing of actions, parallels, mergers, and conditional choices are identified and captured into visual models, in a more comprehensible manner. This identification of process element associations is further utilised in other chapters in determining the impact of changes in these interrelated elements.

3.3.3.3 Process Definition Languages

The third level of process modelling is where process elements and their associations are mapped into machine-readable process modelling languages; for enactment by the workflow engines in WfMS. Similar to visual modelling notations, there are a number of machine-readable process modelling languages. This section briefly acknowledges the prominent process definition languages but does not attempt to discuss their strengths and weaknesses as it is clearly outside the scope of this research. Machine-readable process models are often referred to as Process Definition Languages or Workflow Programming Languages (Weske, 1999). These machine-readable process definitions mostly depend on the implementation of the WfMS.

There have been efforts by WfMC, to standardise process definition languages based on their meta-model. As a result of this standardisation effort, an XML based process definition language named XPDL-XML Process Definition Language was created in 1999 (WfMC). The latest version of XPDL 2.0 (WfMC, 2005) was recently published by WfMC.

XPDL has XML elements that allow high-level process model notations to be captured into machine-readable processes definition languages. These include normal process elements such as actions, participants, data, and business rules; plus other extensions required for WfMS to interact with external systems. Further, XPDL being an XML based, can be extended to suite the implementation of a particular WfMS (WfMC, 2005). Another aspect of XPDL is its strong conformance with the WfMC’s workflow meta-model (previously shown in Figure 3.13). Therefore, any high-level process model that conforms to workflow meta-model can be converted to an XPDL based process definition, for the enactment by the workflow engine.

A number of WfMSs uses XPDL as its underlying process definitions language. Some of the prominent uses of XPDL as mentioned in WfMC’s website are: Fijitsu’s Interstage Suite, IBM’s FileNet Business Process Manager 4.0, and BEA System’s AquaLogic Enterprise Repository and BPM Suite.

Despite the efforts of WfMC to standardise XPDL and its wider usage, there are other process definition languages. A clear competition for XPDL is from
Business Process Execution Languages for Web Services, in short known as BPEL by BEA, IBM and Microsoft (Andrews et al., 2003). BPEL is built on IBM’s WSFL (Web Services Flow Language) and Microsoft’s XLANG (Web Services for Business Process Design) (van der Aalst, Dumas, ter Hofstede, Russell, Verbeek, & Wohed, 2005b).

BPEL, similar to XPDL is an XML based process definition language that is capable of capturing the basic flow controls of a business process. More importantly, the ability to extend process actions using web services allows it to be used for collaborative business processes.

There are a number of criticisms on BPEL, particularly on its ability to model complex flow controls (van der Aalst et al., 2005b; Wohed, van der Aalst, Dumas, & ter Hofstede, 2003). Some researchers have provided MTTs for validating and verifying of BPEL models. For example one method proposed was to transform BPEL based definitions into formalisms such as Petri-Nets (Ouyang, Verbeek, van der Aalst, Breutel, Dumas, & ter Hofstede, 2007; van der Aalst, Dumas, Ouyang, Rozinat, & Verbeek, 2005a). In addition, there are research that study the transformation between BPEL and other high-level process modelling notations such as EPCs (Mendling & Ziemann, 2005b) and WF-Nets (van der Aalst & Lassen, 2005c).

In this research, these process definition languages are analysed in order to understand associations that exists among process elements. These associations give the foundation for further research explained in other chapters.

So far, the state of the art developments in the vast area of business process automation and workflow technologies were presented. Based on this foundation next, issues and challenges in the area of business process and workflow evolutions, and solutions provided for these issues and challenges are discussed in detail.

3.4 Business Process and Workflow Evolution

Despite an abundant number of both commercial and open source WfMS, with their advertised benefits, and standardisation efforts, many organisations still tend to create their own process-oriented applications, which are generally supported by simple workflow engines. Two of the examples familiar to the author are
University of Western Sydney’s and University of Technology Sydney’s (Australia) systems for courses approval named OCAS (2006) and OCAP (2006) respectively. The author believes that readers in the workflow area can relate themselves to being involved in some sort of process-oriented application or WfMS development in their career. This phenomenon raises the question as to why organisations are not embracing the workflow technologies similar to other technologies such as DBMS. Some believe that workflow technology is still at its infancy in comparison to areas such as DBMS (Baeyens, 2004). In this view, over the years many researchers have provided useful guidelines (Hill et al., 2006; van der Aalst, 2002; Stohr et al., 2001; Alonso et al., 1997; Veijalainen et al., 1995) to shape the future of workflow research. When analysing these and many other research studies, the issue of managing business process evolutions can be singled-out as one of the key problems.

Processes of an organisation evolve in order to satisfy legal, strategic, operational and technical reasons (van der Aalst et al., 2000). All these reasons result in either: a) having to introduce progressive alterations, also referred as *evolutions*, to the process, or b) introduce major redesign, referred to as *changes*, in the form of a BPR effort (van der Aalst, Basten, Verbeek, Verkoulen, & Voorhoeve, 1999). These evolutions and changes in business processes are explored in a number of previous research studies. Various terms are used in the literature to denote process changes such as: workflow evolution, workflow adaptation, dynamic workflows, workflow modification, and workflow flexibility (Sadiq, 1999).

This section aims to study previous research closely, with the objective of revealing a lack of work in ‘evolution impact analysis’, which is the focus area of this research.

### 3.4.1 Static and Dynamic Evolution, and Dimensions of Evolution

One of the earliest categorisation of the problem of business process evolutions in WfMS is given by Casati et al. (1998). This categorisation identifies two aspects as static evolution and dynamic evolution.

- Static evolution – refers to challenges and solutions provided in relation to changing workflow models to reflect business process changes.
• Dynamic evolution – refers to issues and solution researched in managing process instances according to changing process models.

Casati (1998) uses the taxonomy suggested in Banerjee et al. (1987), to introduce a formal framework to handle both static and dynamic evolution in workflows. In relation to static evolution, two terms are coined as *declarative and flow primitives* of modification. *Declarative primitives* refer to addition or removal of tasks or variables from the workflow definition. These declarative primitives are similar to extend (addition), reduce (deletion) and replace (mixture of addition and deletion) types of changes identified by van der Aalst et al. (2000). The term *flow primitives* by Casati (1998) refers to changing the ordering of tasks in a process definition. This aspect is re-termed again in 2000 (van der Aalst et al.) as *re-linking*.

From the dynamic evolution point of view, Casati (1998) proposes three methods to handle dynamic evolution in workflow processes as *abort, flush* and *progressive*. *Abort* refers to dropping the old workflow instances and restarting them under the new definition. *Flush* refers to allowing the existing instances to complete using the old process definition. *Progressive* changes refer to adaptation of instances to new process definition based on the status of the instance. Another notable contribution is the concept of introducing evolution policies for handling progressive changes.

While Casati’s (1998) work has laid a solid foundation to both static and dynamic evolution research, there are a number of research studies in workflow evolution, prior and successive to his work. Some of the previous studies under static and dynamic evolution are listed below in their chronological order.

• Ellis et al. (1995) is one of the landmark research studies prior to Casati (1998), which introduces the term “dynamic change bug”, which refers to introducing errors into process instances as a consequence of schema changes.

• Loossens and Van Houtte (1995) take the approach that it is best to know what is likely to change in a process, prior to changes actually taking place, to better manage it. Hence, they propose a mechanism to predict changes based on current and historical data.
• Nutt (1996) describes the state-of-the-art developments in supporting flexibility or evolution by the mid 90s, particularly surveys of the existing approaches such as creating models to manage flexibility with a process.

• Glance, Pagani, and Pareschi (1996) is one of the early works that points to the need for flexible process definition languages. In this view, the authors propose a Generalised Process Structure Grammars (GPSG) that captures constraints and soft dependencies among process actions.

• Vossen et al. (1996) discuss challenges and provides a solution to process evolution in web-based workflow systems.

• Reichert and Dadam (1997) propose an evolution language that consist of a minimal, yet complete set of operations to carryout process changes, based on their ADEPT workflow model.

• Liu, Orlowska, and Li (1998) propose a language for automating the migration (referred to as handover) of running processes instances to the changing process definition.

• Eder and Saringer (1999) suggest the use of hybrid of workflow schemata for handling instances of changing processes.

• Heinl, Horn, Jablonski, Neeb, Stein, and Teschke’s (1999) much cited work introduces two aspects to flexibility in processes: flexibility by selection and adaptation. Flexibility by selection refers to setting multiple paths that can lead to the end goal of the process. Flexibility by adaptation refers to re-inventing a path to reach the end goal of a process. Flexibility by adaptation is similar to static changes, where process definitions are changed according to new needs.

• Sadiq (2000; 1999) discusses the issues and challenges in managing dynamic process and extends the evolution policies by introducing five (5) modifications as: flush, abort, migrate, adapt, and build, to both prevent system and semantic failures.

• van der Aalst et al. (2001a; 1999) provide greater insight to the problem of handling both static and dynamic evolutions, with the importance of addressing the problem in different dimensions such as process actions, participants, and
objects (referred to as process case). However, the solutions are primarily provided for handling evolutions of process actions.

- Kradolfer and Geppert (1999) propose an interesting approach for managing dynamic evolution by versioning the process definitions and migrating the instances based on migration rules.

- Kammer, Bolcer, Taylor, Hitomi, and Bergman (2000) provide further insight into the problem of handling process instances with changing process definitions. In this work they highlight that changes affect instances in different ways and there is a need for a multitude of approaches for handling them, within the same WfMS.


- Narendra (2004; 2000) discusses the classic issue of keeping the balance between flexibility required by changing processes and control required by process managers. In this view an architecture that is suggested sufficiently handles both aspects of the problem.

- Weske (2001a) proposes a formal foundation for handling dynamic changes in an object oriented environment.

- In the area of static evolution, Sadiq and Orlowska (2001) propose a new definition for the term flexibility as the ability of the workflow process to execute on the basis of a loosely, or partially specified model, where the full specification of the model is made at runtime, and is unique to each instance. To facilitate this, they introduce the notion of an open instance that consists of a core process and several pockets of flexibility.

- Carrying out the dynamic changes in workflow processes is one aspect, however evaluating and ensuring that changes are correct is another issue. In this view, Rinderle et al. (2003) propose a criteria for evaluating the accuracy of dynamic changes based on post-change behaviour. These criteria include concepts such as completeness, correctness and change realisation.

- Domingos and Veiga (2003) discuss the challenges in relation to managing process participants perspective as the static and dynamic evolution takes place.
Then they propose a mechanism based on RBAC (role based access control) to handle adaptive workflows.

- Rinderle et al. (2004) discuss the issues of adaptation and flexibility from a collaborative business processes point of view. In this work they present general and comprehensive correctness criteria for ensuring compliance of in-progress WF instances with a modified WF schema.

- Rinderle et al. (2005) is one of the research studies that identifies the need for a holistic solution to process evolution, taking into consideration a process lifecycle. In their approach they suggest to record history of changes giving the ability to have traceability of changes for future successful evolutions.

- Lu et al. (2007; 2006b) and Lu and Sadiq (2006a) provide the concept of using successful previous work practices for the purpose of modelling flexible business processes. For this purpose, they utilise the method of adaptations effectively for process improvement through effective management of the process variants repository (PVR).

The above list is only a representative sample of the vast amount of literature relevant to this research. When analysing the above list, what is interesting to notice is that workflow evolution is a topic that has been researched over a decade, providing islands of solutions. However, to date many organisations still struggle to manage their automated processes with changing requirements and needing to work-around-the-systems to find solutions. The author believes one reason behind this issue is the lack of holistic approaches that can facilitate real business issues in relation to process evolution.

In relation to providing a holistic solution to workflow evolution, van der Aalst and Jablonski (2000), introduce three dimensions: resource similar to process participants, process referring to actions and their ordering and cases denoting the process object (see Figure 3.21).
Based on these three dimensions, van der Aalst and Jablonski (2000) also present a set of criteria for handling and managing workflow evolution. While this set of criteria promises to provide a holistic solution, more importance is given only in relation to process action changes, and evolution of other dimensions are abandoned.

The need for addressing the issue of workflow evolution beyond the workflow actions dimension is emphasised by other researchers as well. For example, Sadiq et al. (2004) and Zhang and Wang (2005) have pointed out the need for handling the evolutionary nature of a process object and its associated data. This relates to the practical situation, where in businesses, forms and other production documents are constantly changed due to various reasons. However, workflow and in particular workflow evolution research, has largely neglected the changes that could take place in process related information objects and its follow on effect on the workflow.

The other aspect, which is inadequately researched, is the evolutionary nature of organisational structures and its subsequent effect on the processes. While there are some research (Sun & Pan, 2005; Domingos et al., 2003; Ahn et al., 2000) that attempt to address the issues with role based access control mechanisms, the core problem of organisational changes and its impact on process models are yet another area that needs further research.

The above discussion relates to the background on workflow evolution. The next section discusses some advanced topics, in relation to workflow evolution. These topics are workflow analysis (termed by van der Aalst et al. (2003c)) and propagating impact analysis (termed by Bodhuin et al. (2004)).
3.4.2 Workflow Analysis and Propagating Impact Analysis

Making changes to process definitions (static evolution) or migrating running instances (dynamic evolution) are only one part of the workflow evolution management. Closely associated with workflow evolution is the issue of assuring that inconsistencies or errors in process models are not created, due to changes. This aspect, of confirming the accuracy of changed process models is referred to as workflow analysis (van der Aalst, 1998).

In relation to assuring the correctness of process models, three analytical concepts have been introduced. These concepts are process verification, performance analysis, and validation (Sheth, Van Der Aalst, & Arpinar, 1999; van der Aalst, 1998).

The term verification refers to assuring the syntactical or structural correctness of the process models. The issue of verification predominantly depends on the process modelling tools used. For instance, the syntactical correctness of WF-nets and BPMN models cannot be resolved unless both these models are converted into a single platform such as Petri-Nets. An example is conversion of UML based process models to Petri-Nets for analytical purposes (van der Aalst et al., 2003a).

The verification of process models is implemented in varying degrees in WfMS. This is mainly due to the fact that verification methods depend on the modelling tool and its analytical capacity (van der Aalst et al., 2003c). On some occasions, there are tools that are purely intended for verification of process models. Some examples of such tools are Woflan (Verbeek, Basten, & van der Aalst, 2001; Verbeek & van der Aalst, 2000) a process model verification tool based on Petri-Nets, UML activity diagram verification tool (Eshuis et al., 2004) and Yasper (van Hee et al., 2006) another Petri-Nets based verification tool.

The performance evaluation refers to finding the throughput and bottlenecks associated with process models. According to van der Aalst et al. (2003c) not many WfMS provide facilities for process performance analysis of a modelled process. However, there are a number of research studies (Aiello, 2004; Li, Fan, & Zhou, 2004; McGregor, 2002) that provides MTTs in relation to performance evaluation.

Making certain that process models accurately reflect the intended organisational business needs is referred to as validation (Shelomanov, 2003; van
der Aalst, 2002). Simulation is the widely used mechanism, which is supported in some WfMS (van der Aalst et al., 2003c). Process simulation is the technique that allows representation of processes, people, and technology in a dynamic computer model (Tumay, 1996). The success of simulation depends on a number of factors:

- **The simulation capability of the tool** – There are a number of simulation approaches and tools that use a selected approach (Tumay, 1996). Jansen-Vullers and Netjes (2006) survey a number of process simulation tools, to reveal the strengths and weaknesses of them. According to Jansen-Vullers and Netjes (2006) no one tool provides all the capabilities expected in process simulation.

- **The data set used for simulation** – The success of simulation depends on the data set used for simulation purposes. Creating a proper data set that captures all possible process scenarios can be both time consuming and costly (Bansal, 2002). The use of ‘dummy-data’ creates the situation of ‘garbage-in-garbage-out’, giving false analytical information.

- **Experience and expertise of the simulation specialists** – Another issue with simulation is the experience and expertise of process managers or business analysts who analyse the simulation data to predict the improvement possibilities (Bansal, 2002). Generally, it is cost effective to use junior staff who are less experienced for such tasks. This is with the risk of missing some crucial and costly inconsistencies that would be realised only during the real production time.

Based on the above-mentioned reasons, the author believes that in process evolution, simulation alone is not sufficient for assuring that changes are error free and in accordance with organisational needs. The requirement here is for repositories that facilitate providing guided support, for business analysts and process managers to make process changes confidently. For this purpose, it is vital to have proper knowledge on propagating impact of a change.

The need for propagating impact analysis has been recently highlighted by other researchers (Ramesh et al., 2005; Soffer, 2005; Bodhuin et al., 2004) as well. The author defines the propagating impact to be the ripple or domino effect created, when a change is introduced to a process. Unless impact analysis is adequately
handled, the time and costs involved in reflecting the corresponding process changes on the already implemented processes can significantly increase.

*Workflow analysis*, which consists of techniques such as *validation, verification, and performance analysis* and *propagating impact analysis*, is considered vital for successful evolution management of process-oriented applications. In this section, a number of research studies in the area of workflow analysis are explored in detail in order to find the analytical aspects that they cover. The objective of this exploration is to find the depth of coverage of the following aspects in published research:

- **The nature of Workflow Analysis Supported** – This is to categorise the core analytical concepts being covered. These concepts would include previously discussed *validation, verification, or performance analysis*.

- **Workflow modelling language/notations analysed** – Most of the workflow analytical research focuses only on one process-modelling tool or language. This criterion lists the modelling tool or language upon which the analysis is performed.

- **Underlying Formalism/ Theoretical approach used for analytical purposes** – This represents the different analytical methods deployed for analysing process models. These methods could include Petri-Net based algorithms, State-chart based analytical methods, probability distribution method based analytical methods (Jansen, 2002), queuing theory based algorithms or custom analytical algorithms.

- **Supported Process Elements** – Not all research attempts to support the analysis of the main three perspectives of a process namely activities, participants, and data object. These criteria therefore looks at the process perspective/s supported.

- **Facility to find the propagating impact of a change** – The last criterion is to check whether this analytical method can help in finding the propagating impact of process element changes.

The comparison of previous research against above set of criteria is presented in tabular format in Table 3.1. The literature presented in Table 3.1 is in the
chronological order and closely related research studies are analysed together. In presenting the research findings, several notations are used in this table:

- **Plus sign (+)** indicates the aspect is fully supported without requiring any modifications,
- **Less than sign (<)** indicates that some support is provided (under certain conditions) in the current form but not fully supported,
- **Greater than (>)** sign indicates that support can be given with certain known extensions,
- **Minus sign (-)** denotes that support is not given in the current form with known modifications or extensions,
- **Question mark (?)** denotes that this aspect is identified to be as required but fails to provide sufficient details or the information is unclear, and
- **Forward slash (/)** denotes that the aspect does not fall within the scope of that research.
Table 3.1- Summary of Literature in the area of Workflow Analysis and Propagating Impact Analysis

Notations used: (+) Fully supported without requiring any modifications, (<) Some support is provided in the current form, (>) With certain known extensions some support can be provided, (-) Not supported in the current form with known modifications or extensions, (?) Though mentioned, enough details not given or unclear. (/) Does not fall within the scope of the research.

<table>
<thead>
<tr>
<th>Previous Research Work</th>
<th>The nature of Workflow Analysis Supported</th>
<th>Workflow modelling language/notations analysed</th>
<th>Underlying Formalism/Theoretical approach used for analytical purposes</th>
<th>Supported Process Elements</th>
<th>Facility to find the propagating impact of a change</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. (Adam, Atluri, &amp; Huang, 1998)</td>
<td>Verification and Validation</td>
<td>Workflow model introduced in (Georgakopoulos et al., 1995)</td>
<td>Petri-Nets based approach</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>2. (Castano, De Antonellis, &amp; Melchiori, 1999)</td>
<td>Analysis of Re-Engineering needs of a process</td>
<td>Custom workflow model</td>
<td>By analysing the inputs and outputs sent to an action</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>3. (van der Aalst, Hirnschall, &amp; Verbeek, 2002b; Verbeek et al., 2001; Sadiq et al., 2000b; Sadiq &amp; Orlowska, 1999a)</td>
<td>Verification of structural correctness</td>
<td>WF-Graphs</td>
<td>A combination of Petri-Net analysis techniques or graph reduction method.</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>4. (Marjanovic, 2000)</td>
<td>Verification of temporal consistency</td>
<td>Extended WF-Graphs</td>
<td>Visual verification based on models</td>
<td>+</td>
<td>-</td>
</tr>
</tbody>
</table>
### Notations used:

- (+) Fully supported without requiring any modifications,
- (<) Some support is provided in the current form,
- (> With certain known extensions some support can be provided,
- (-) Not supported in the current form with known modifications or extensions,
- (?) Though mentioned, enough details not given or unclear.
- (/) Does not fall within the scope of the research.

<table>
<thead>
<tr>
<th>Previous Research Work</th>
<th>The nature of Workflow Analysis Supported</th>
<th>Workflow modelling language/notations analysed</th>
<th>Underlying Formalism/Theoretical approach used for analytical purposes</th>
<th>Supported Process Elements</th>
<th>Facility to find the propagating impact of a change</th>
</tr>
</thead>
<tbody>
<tr>
<td>7. (Beydeda et al., 2001)</td>
<td>Risk analysis of change</td>
<td>Custom workflow model consisting of actions edges</td>
<td>Custom algorithm that searches the model based on activity time</td>
<td>(+) (-) (-) (-)</td>
<td>(&lt;)</td>
</tr>
<tr>
<td>9. (Eshuis et al., 2004, 2002; Eshuis, 2002)</td>
<td>Verification of syntactic and structural correctness</td>
<td>UML activity diagram based process models</td>
<td>Activity Hypergraphs combined with Linear Temporal Logic</td>
<td>(+) (-) (-) (-)</td>
<td>(?)</td>
</tr>
<tr>
<td>10. (Sivaraman et al., 2005; Sivaraman, 2003)</td>
<td>Verification and Validation</td>
<td>Control flow model introduced in (Sivaraman et al., 2005)</td>
<td>Petri-Net based algorithm named KORRECTNESS Algorithm (Sivaraman, 2003)</td>
<td>(+) (-) (-) (-)</td>
<td>(?)</td>
</tr>
<tr>
<td>11. (Sadiq et al., 2005; Sadiq et al., 2004)</td>
<td>Validation of process models</td>
<td>Directed Acyclic Graph (Sadiq et al., 1997)</td>
<td>Custom algorithm</td>
<td>(+) (-) (-) (-)</td>
<td>/</td>
</tr>
<tr>
<td>12. (Bodhuin et al., 2004)</td>
<td>Change impact analysis for co-evolution of processes and systems</td>
<td>UML based process models are used for examples</td>
<td>+ (-) (-) (-)</td>
<td>/</td>
<td></td>
</tr>
<tr>
<td>13. (Li, Yang, &amp; Chen, 2004)</td>
<td>Verification of resource consistency</td>
<td>Custom workflow model introduced in the work.</td>
<td>Custom algorithm</td>
<td>(&lt;) (+) (+)</td>
<td>/</td>
</tr>
</tbody>
</table>
### Notations used:
- (+) Fully supported without requiring any modifications,
- (<) Some support is provided in the current form,
- (> With certain known extensions some support can be provided,
- (-) Not supported in the current form with known modifications or extensions,
- (?) Though mentioned, enough details not given or unclear.
- (/) Does not fall within the scope of the research.

<table>
<thead>
<tr>
<th>Previous Research Work</th>
<th>The nature of Workflow Analysis Supported</th>
<th>Workflow modelling language/notations analysed</th>
<th>Underlying Formalism/Theoretical approach used for analytical purposes</th>
<th>Supported Process Elements</th>
<th>Facility to find the propagating impact of a change</th>
</tr>
</thead>
<tbody>
<tr>
<td>14. (Watanabe &amp; Kaneda, 2004)</td>
<td>Performance analysis to remove ineffective process actions</td>
<td>?</td>
<td>Activity based costing to find non-value adding actions and repetition of value adding actions</td>
<td>+</td>
<td>&lt;</td>
</tr>
<tr>
<td>15. (Fu, Bultan, &amp; Su, 2004)</td>
<td>Validation</td>
<td>BPEL</td>
<td>Using Promela (a modelling language for finite-state concurrent processes) and related analytical algorithms. BPEL models are translated to BPEL to Promela</td>
<td>+</td>
<td>&gt;</td>
</tr>
<tr>
<td>17. (Ouyang et al., 2007; Verbeek &amp; van der Aalst, 2005)</td>
<td>Verification of process models</td>
<td>BPEL</td>
<td>Converts the BPEL elements to Petri-Nets prior to using Petri-Net based algorithm</td>
<td>+</td>
<td>-</td>
</tr>
</tbody>
</table>
### Notations used:
- (*) Fully supported without requiring any modifications,
- (<) Some support is provided in the current form,
- (>)* With certain known extensions some support can be provided,
- (-) Not supported in the current form with known modifications or extensions,
- (?)* Though mentioned, enough details not given or unclear.
- (/) Does not fall within the scope of the research.

<table>
<thead>
<tr>
<th>Previous Research Work</th>
<th>The nature of Workflow Analysis Supported</th>
<th>Workflow modelling language/notations analysed</th>
<th>Underlying Formalism/Theoretical approach used for analytical purposes</th>
<th>Supported Process Elements</th>
<th>Facility to find the propagating impact of a change</th>
</tr>
</thead>
<tbody>
<tr>
<td>19. (Hsu, Wang, &amp; Yang, 2005)</td>
<td>Verification of process models</td>
<td>Directed Acyclic Graph (Sadiq et al., 1997) an Extended version</td>
<td>A set of custom designed incremental algorithms</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>20. (Ramesh et al., 2005)</td>
<td>Change Impact analysis - proposes a mechanism to identify the impact of context evolution that leads to process redesign.</td>
<td>?</td>
<td>Knowledge base that manages the associations of process models and querying it</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>21. (Stefanov et al., 2005)</td>
<td>Performance analysis of business processes</td>
<td>Event-driven Process Chains</td>
<td>Using a data warehouse to record the performance measurement values and querying it</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>22. (Liu &amp; Kumar, 2005)</td>
<td>Verification of models un-structured processes</td>
<td>Directed graphs</td>
<td>Identifying the patterns of possible structural flaws and searching for their existence to identify the issues. Then suggesting correct structure to rectify the problem</td>
<td>+</td>
<td>-</td>
</tr>
</tbody>
</table>
Notations used: (+) Fully supported without requiring any modifications, (<) Some support is provided in the current form, (> With certain known extensions some support can be provided, (-) Not supported in the current form with known modifications or extensions, (?) Though mentioned, enough details not given or unclear. (/) Does not fall within the scope of the research.

<table>
<thead>
<tr>
<th>Previous Research Work</th>
<th>The nature of Workflow Analysis Supported</th>
<th>Workflow modelling language/notations analysed</th>
<th>Underlying Formalism/ Theoretical approach used for analytical purposes</th>
<th>Supported Process Elements</th>
<th>Facility to find the propagating impact of a change</th>
</tr>
</thead>
<tbody>
<tr>
<td>23. (Soffer, 2005)</td>
<td>Finding the scope of the change to either no impact, local impact or global impact referring to other workflows in the organisation</td>
<td>Custom workflow model</td>
<td>Custom analytical algorithm</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>24. (van Hee et al., 2006)</td>
<td>Simulation of processes using visual tool for validation</td>
<td>WF-net based process models</td>
<td>Petri-Nets based algorithms</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>25. (Tian, Xing, &amp; Li, 2006)</td>
<td>Performance analysis</td>
<td>WF-Nets</td>
<td>Custom algorithm, which analysis the time constraints of process activities using both fuzzy and statistical temporal information.</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>26. (Ray &amp; Xin, 2006)</td>
<td>Validation</td>
<td>Advance Transactional Model proposed in (Ray et al., 2006)</td>
<td>Custom algorithm that determines the dependencies among transactions</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>27. (Vanderfeesten, Reijers, &amp; van der Aalst, 2007)</td>
<td>Measuring the quality workflow by getting the best suited level of cohesion and coupling of process activities</td>
<td>Petri-Nets based directed graphs</td>
<td>Using heuristic methods</td>
<td>+</td>
<td>&gt;</td>
</tr>
</tbody>
</table>
When analysing the above sample set (over 35) of workflow analysis literature presented in Table 3.1, it is evident that most research focuses on verification and validation of process models.

Close to 50% of the works concentrate on verification of a number of process modelling mechanisms such as UML activity diagrams, Wf-Nets, WF-graphs, BPEL processes, or other custom workflow models. For verification a multitude of mechanisms are used from visual inspection, algorithmic methods to mathematical formalisms. In the case when a process-modelling tool does not directly support visual representations or formalisms (for example BPEL), the models are converted to formalism such as Petri-Nets for verification.

About 20% of the workflow analytical research studies concentrate on answering the question ‘have we built the right process?’ or in other words validation (Desel, 2002). The common mechanisms used for validation seems to be either visual inspection or simulation. However, as mentioned earlier, there are certain limitations in relation to simulation that could leave inconsistencies and errors undetected in the process models. The success of visual inspection is limited to relatively smaller processes and the analytical skill of the visual inspector.

Close to 15% of the workflow analytical research (six in total number), had some association with propagating impact analysis, which is the core focus of this research. However, the author is certain that this 15% covers almost all the research in relation to change impact analysis.

The issues with the research in the area of impact analysis were the lack of information on the approaches used and the incompleteness of their approaches. For instance, some studies (Bodhuin et al., 2004; Castano et al., 1999) that discuss the issue of impact analysis do not provide an approach or solution to the problem or their approaches are unclear. Research by Nissen et al. (2000), Beydeda and Gruhn (2001), Ramesh et al. (2005) and Soffer (2005) only concentrate on process actions and participants; and similar to most other workflow analysis research neglects the process data object aspect.

About 10% of above listed studies concentrate on performance analysis of models. However, the author acknowledges that work related to performance analysis was analysed sparsely, due to its little or no relevance. In addition, about 5%
of work looks at other issues such as quality assurance, which are clearly beyond the scope of this research.

This lack of research and holistic approaches in the area of impact analysis, coupled with the real-life problem in relation to the OCAS (2006) project (details are discussed in Chapter-4) are the core motivations for this research. The solution, for the problem of propagating impact analysis is systematically developed in the subsequent chapters of this thesis. In providing this solution, there are a number of theoretical concepts used. The next section presents an overview of these theoretical concepts used in providing the solution to the issue of propagating impact analysis.

3.5 Solution Space

The in-depth analysis in the previous section shows an obvious gap in the area of change impact analysis in relation to process evolution in workflow systems, in particular in web-based workflows. The subsequent chapters build up a solution to the problem, identified by this gap. In arriving at a solution, a number of existing theoretical concepts are used.

This section aims at exploring a candidate set of MTTs that could be used for this purpose. The broader methodology or approach used is to comprehend all possible associations of process elements not only among themselves, but also with other elements such as with contextual artefacts, models and web-based workflow systems artefacts. The technique planned to be used is to capture these associations into a structure that can be searched or analysed to find the propagating impact. The possible tools are relational data structures, binary trees, formalisms, and searching or analytical algorithms.

The rest of this section investigates the current state of these theoretical concepts in relation to BPA and BPEM.

3.5.1 Associations of Process Elements

The solution provided for change impact analysis of processes is mainly founded on comprehension and abstraction of dependencies, association, and constraints of real-life process. The importance of comprehension of such associations, prior to propagating impact analysis was first introduced by Bodhuin et
al. (2004). These associations are then used to trace the propagating impacts of changes that initiate at: contextual artefacts in the pragmatic level, real-life processes in the semantic level, models in the syntactic level, and implemented artefacts in the implementation level (van der Aalst et al., 2000).

This section therefore investigates the existing MTTs available for capturing a number of associations such as, a) associations among process elements, b) process element associations with contextual artefacts, c) process element association with abstract models, and d) process element association with workflow artefacts.

- **Associations among process elements** – Each process element is associated with other process elements in order to achieve the business goals of the process. These associations among process elements can be further categorised into six types: among actions, among participants, among object data, between action and participants, between action and data, and between data and participants.

- **Process element associations with contextual artefacts** – Contextual artefacts generally define the existence of a process element and the above-mentioned association with other process elements.

- **Process element association with abstract models** – Once the processes are identified for automation models are created. This aspect suggests the associations of process elements with models.

- **Process element association with workflow artefacts** – Workflow systems automate the business process; thus, every process element that is automated has an association with an implementation artefact of the WfMS.

Using the above types of associations as the evaluation criteria, previous research studies are analysed, to comprehend their strengths and weaknesses in conceptualising associations of process elements. This information is presented in Table 3.2.

In this table, textual descriptions are used where appropriate; and notations +, -, <, >, ?, and / are also used, similar to Table 3.1. These previous literature is recorded in the chronological order in Table 3.2 and related studies are grouped together.
Table 3.2- Strengths and Weaknesses of Process Conceptualisation Models in Capturing Associations of Process Elements

<table>
<thead>
<tr>
<th>Description / Name</th>
<th>Among process elements</th>
<th>Associations supported</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>process elements and contextual information</td>
</tr>
<tr>
<td>1. (Eder et al., 1995; Eder &amp; Liebhart, 1994)</td>
<td>Workflow Activity Model (WAMO) that allows modelling process and known exceptions</td>
<td>+ among actions &lt; among participants - among object data + between action and participants - between action and object data - between data and participants</td>
</tr>
<tr>
<td>2. (Kappel, Rausch-Schott, &amp; Retschitzegger, 2000; Gottlob, Schrefl, &amp; Rock, 1996; Kappel, Lang, Rausch-Schott, &amp; Retschitzegger, 1995)</td>
<td>Object-Oriented approach to capture the association between process elements and in particular use of ECA (Event Constraint and Action) rule based approach for activity coordination</td>
<td>&lt; among actions + among participants + among object data + between action and participants + between action and object data ? between data and participants</td>
</tr>
<tr>
<td>3. (Bussler et al., 1995)</td>
<td>A framework for handling the associations between process elements and contextual artefacts in the form of policies and organisational structures</td>
<td>/ among actions + among participants / among object data &gt; between action and participants / between action and object data / between data and participants</td>
</tr>
</tbody>
</table>

Notations used: (+) Fully supported without requiring any modifications, (<) Some support is provided in the current form, (>) With certain known extensions some support can be provided, (-) Not supported in the current form with known modifications or extensions, (?) Though mentioned, enough details not given or unclear, (/) Does not fall within the scope of the research.
**Notations used:** (+) Fully supported without requiring any modifications, (<) Some support is provided in the current form, (>) With certain known extensions some support can be provided, (-) Not supported in the current form with known modifications or extensions, (?) Though mentioned, enough details not given or unclear. (/) Does not fall within the scope of the research.

<table>
<thead>
<tr>
<th>Description / Name</th>
<th>Associations supported</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Among process elements</strong></td>
<td><strong>process elements and contextual information</strong></td>
</tr>
<tr>
<td>4. (Kwan et al., 1997) Meta-model named Dynamic Workflow Management (DWM) Meta-Model</td>
<td>+ among actions</td>
</tr>
<tr>
<td></td>
<td>+ among participants</td>
</tr>
<tr>
<td></td>
<td>+ among object data</td>
</tr>
<tr>
<td></td>
<td>+ between action and participants</td>
</tr>
<tr>
<td></td>
<td>+ between action and object data</td>
</tr>
<tr>
<td></td>
<td>+ between data and participants</td>
</tr>
<tr>
<td>5. (Lu et al., 2007; Mangan et al., 2002b; Mangan et al., 2002a; Sadiq &amp; Orłowska, 1999b; Sadiq et al., 1997) Directed graphs for showing the association among process elements and using constraint specification for capturing other associations. Uses an object-relational data structure to support the constraints.</td>
<td>+ among actions</td>
</tr>
<tr>
<td></td>
<td>- among participants</td>
</tr>
<tr>
<td></td>
<td>- among object data</td>
</tr>
<tr>
<td></td>
<td>+ between action and participants</td>
</tr>
<tr>
<td></td>
<td>+ between action and data</td>
</tr>
<tr>
<td></td>
<td>? between data and participants</td>
</tr>
<tr>
<td>6. (Russell, van der Aalst, &amp; ter Hofstede, 2006b; Russell et al., 2006a; Russell, ter Hofstede, Edmond, &amp; van der Aalst, 2005a; Russell et al., 2005b; van der Aalst et al., 2003b; van der Aalst, 2003b; van der Aalst, 1998, 1997) This series of work studies the common patterns of process element association. This work is on WF-nets a Petri-Net based visual model</td>
<td>+ among actions</td>
</tr>
<tr>
<td></td>
<td>- among participants</td>
</tr>
<tr>
<td></td>
<td>- among object data</td>
</tr>
<tr>
<td></td>
<td>+ between action and participants</td>
</tr>
<tr>
<td></td>
<td>+ between action and data</td>
</tr>
<tr>
<td></td>
<td>- between data and participants</td>
</tr>
<tr>
<td>Description / Name</td>
<td>Associations supported</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>Workflow meta-model proposed by WfMC</td>
<td>- among actions</td>
</tr>
<tr>
<td>Proposes a standard for identifying associations that process participants have among themselves and with actions and object data. This mainly for access control purposes</td>
<td>- among actions</td>
</tr>
<tr>
<td>UML based patterns for identifying common process patterns in administrative business processes</td>
<td>+ among actions</td>
</tr>
<tr>
<td>This is a meta-model, and does not exclusively support any particular modelling notation. However applicability is shown with ECPs and indicates the possibility of using with BPMN</td>
<td>+ among actions</td>
</tr>
</tbody>
</table>

**Notations used:** (+) Fully supported without requiring any modifications, (<) Some support is provided in the current form, (>) With certain known extensions some support can be provided, (-) Not supported in the current form with known modifications or extensions, (?) Though mentioned, enough details not given or unclear. (I) Does not fall within the scope of the research.
<table>
<thead>
<tr>
<th>No.</th>
<th>Source/Year</th>
<th>Description / Name</th>
<th>Associations supported</th>
<th>Additional information</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.</td>
<td>(van der Aalst et al., 2005d; van der Aalst, 2001b)</td>
<td>A meta-model and a formalism for using case handling approach for process modelling</td>
<td>+ among actions</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&gt; among participants</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+ among object data</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+ between action and participants</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+ between action and object data</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&gt; between data and participants</td>
<td>/</td>
</tr>
<tr>
<td>12.</td>
<td>(Ramesh et al., 2005)</td>
<td>Proposes need for associating process elements with contextual artefacts for the</td>
<td>? among actions</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td></td>
<td>purpose of finding propagating change impact</td>
<td>? among participants</td>
<td>&lt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- among object data</td>
<td>&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>? between action and participants</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- between action and data</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>? between data and participants</td>
<td>/</td>
</tr>
<tr>
<td>13.</td>
<td>(Stefanov et al., 2005)</td>
<td>Shows the possibility of EPC (Event-driven Process Chains) to show the association</td>
<td>+ among actions</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>between different process elements</td>
<td>? among participants</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>? among object data</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+ between action and participants</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+ between action and data</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+ between data and participants</td>
<td>/</td>
</tr>
<tr>
<td>14.</td>
<td>(Vanderfeesten et al., 2007)</td>
<td>Petri-Nets based direct graph method</td>
<td>+ among actions</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- among participants</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- among object data</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+ between action and participants</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>+ between action and data</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- between data and participants</td>
<td>/</td>
</tr>
</tbody>
</table>
According to the literature presented in Table 3.2, only one conceptualisation mechanism (Kwan et al., 1997) supports capturing all six types of associations among process elements. A few other researchers (Stefanov et al., 2005; van der Aalst et al., 2005d; van der Aalst, 2001b; Kappel et al., 1995) have mentioned the need for understanding all these six types of associations, however they fail to explicitly provide them in their solutions.

In relation to associations that process elements have with contextual artefacts is only recognised in the works of Ramesh et al. (2005), Zur Muehlen (2004c), and Bussler and Jablonski (1995). However, it should be worth mentioning that the associations of contextual artefacts have been outside the scope of most research.

The other two types of associations that link process elements with conceptual models and implementation artefacts are least covered. Similar to contextual artefacts, these associations are mostly considered as outside the scope in most literature.

When analysing the above it can be seen a model close to Kwan et al.’s (1997) DWM meta-model is best suited for capturing associations among process elements. However, Kwan et al.’s (1997) model does not support other associations, such as process element associations with contextual artefacts or implementation components of the web-application.

### 3.5.2 Formalisms and Data Structures

The previous section identified a number of associations to be captured, for identifying propagating impact of process element changes. Comprehension of the existence of such associations is only one part of the problem. The other issue is capturing such complex correlations into a data structure that can be implemented for analysis. In this respect, two techniques could be used for capturing the aforementioned associations: *formalisms* that facilitate mathematical analysis and *data structures* similar to *relational databases* and *binary trees*.

#### 3.5.2.1 Formalisms

The capturing of associations among process elements for analytical purposes uses formalisms, such as Petri-Nets. This is evident when examining the findings
presented in Table 3.1. For example, process models in BPEL are converted to Petri-Nets based models for analytical purposes. Apart from Petri-Nets, what other formalisms are available for representing complex associations among process elements? To find answers to this question, the author has selected two other formalisms to be studied along with Petri-Nets, to evaluate their usage particularly in the workflow domain. These two formalisms are *Process Algebra*, and *Kleene Algebra with Tests (KAT)*.

As mentioned earlier Petri-Nets are graphical tools that are directed, weighted and bi-partite (van der Aalst, 2003, 2002). The classical Petri-Nets which were introduced by Carl Adam Petri (1962), consist of places, transitions and arcs (Working-Group, 1997). The *transitions* can be used to represent a variety of concepts such as events, actions, computation steps, tasks, or jobs. The *places* can be used to show pre-conditions, post-conditions, input and output data, input, and output signals, resources needed and conclusions. Using the directed arcs that connect places and transitions, the process models can be defined, such as in WF-nets (van der Aalst, 1999, 1998, 1997).

Graphical representation of Petri-Nets is considered desirable for humans to comprehend process models. However, the use of such models for computer manipulation, such as comparing similarity or differences in processes, and propagating impact analysis, is limited (van der Aalst, 2003). In relation to such computational analysis, the representation of processes as linear expressions is much more advantageous.

In creating linear expressions for computational purposes formalisms such as Process Algebra is useful (Camara, Canal, Cubo, & Vallecillo, 2005). Process Algebra consists of algebraic language for the specification of processes and tools for formulation of statements about the processes, plus calculi for the verification of these statements (van Glabbeek, 1987). Process Algebra has constructs for representing sequential composition, parallels, choices and recursive actions as follows (Amici, Cacciagrano, Corradini, & Merelli, 2004):

- Sequence of two actions A followed by B $\rightarrow$ A;B
- Parallel composition of two actions A and B $\rightarrow$ A||B
- Choice between two actions A and B $\rightarrow$ A □ B
• Recursion of action $A \rightarrow \mu xA$

Using these constructs it is possible to formulate process expressions, which could be used for analytical purposes (van Glabbeek, 1987).

The major difference between Process Algebra and Petri-Nets (apart from the graphical and textual representation) is that Process Algebra is uni-partite. This uni-partite nature is less expressive when considering the rich structure that Petri-Nets have with places and transitions.

Kleene Algebra with Tests (KAT) (Kozen, 1999), which is an extension of Kleene Algebra by Kleene (1956). Similar to Process Algebra, with KAT linear expressions can be created for computational purposes. In addition, KAT allows bi-partite representation. In other words, KAT is a two-sorted algebraic structure $(B, K, +, \cdot, *, 0, 1, \sim)$ where:

- $B$ is a subset in $K$
- $+$ is a binary operator similar to conditional choice (OR operator) for elements from both $K$ and $B$
- $\cdot$ is a binary operator similar to sequence (or AND) operator for elements from both $K$ and $B$
- $*$ is a unary operator, similar to recursion acting only on elements from $K$
- $\sim$ is a unary operator, similar to negation, defined only on $B$
- $0$ and $1$ special operators are similar to null and skip operators

According to above, $(K, +, \cdot, *, 0, 1)$ is a Kleene algebra and $(B, +, \cdot, \sim, 0, 1)$ is a Boolean algebra (Kozen, 1997)

The elements from $B$ in KAT are similar to places in Petri-Nets that can be used to represent a multitude of concepts such as pre-conditions, post-conditions input and output data, input and output signals, resources needed and conclusions. Elements from $K$ are similar to transitions in Petri-Nets, which can represent events, actions, computation steps, tasks, or jobs. These two types of elements allow representing richer information in comparisons to only single type of elements in Process Algebra. Further, KAT can be used to create linear expressions of process descriptions allowing it to be used for computer manipulation.

In Table 3.3 expressive power of Petri-Nets, Process Algebra and KAT are compared. The aim of this comparison it to evaluate their suitability in capturing the associations described in Table 3.2.
Similar to previous Tables (3.1 and 3.2) the notations +, -, <, >, ?, and / are used to present the findings. The literature used to collate the information within the table is noted alongside the column headings- Petri-Nets, Process Algebra and Kleene Algebra with Tests (KAT).

Table 3.3- Capability of Formalisms to Represent Correlations of Process Elements

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Within actions</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Within participants</td>
<td>&gt;</td>
<td>&lt;</td>
<td>&gt;</td>
</tr>
<tr>
<td>Within object data</td>
<td>&gt;</td>
<td>&lt;</td>
<td>&gt;</td>
</tr>
<tr>
<td>Between action and participants</td>
<td>&gt;</td>
<td>&lt;</td>
<td>&gt;</td>
</tr>
<tr>
<td>Between action and data</td>
<td>&gt;</td>
<td>&lt;</td>
<td>&gt;</td>
</tr>
<tr>
<td>Between data and participants</td>
<td>&gt;</td>
<td>&lt;</td>
<td>&gt;</td>
</tr>
<tr>
<td>With Process Elements with Contextual Artefacts</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Process Elements with Modelling Artefacts</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Process elements with Implementation Artefacts</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

When analysing the findings in the above table, it is clear that due to bipartite nature, both Petri-Nets and KAT can be used to capture more associations in comparison to Process Algebra. In addition, it is noticeable in Table 3.3 that process element associations with external elements such as contextual, modelling and implementation artefacts cannot be represented using any of the formalisms. This gives the need for combining the formalisms with a structure that is suitable for capturing these other associations.

With this understanding of the need for leaner expressions to capture process associations and its lack of support to link with other external artefacts, next appropriate data structures are sought to implement these expressions.
3.5.2.2 Relational Data Structures and Binary Trees

In the implementation of associations among process elements in WfMS, generally it is advocated to use machine-readable process definition languages such as XPDL or BPEL. However, when the same associations need to be captured for analytical purposes, such as finding propagating impact, the use of verbose machine-readable process expressions is not suitable. This gives the requirement for a suitable data structure that can be used to capture complex associations among process elements, yet provide easy computer manipulation. In this view, there are two possible candidates: Relational Data Structures and Binary Trees.

A relational data structure is one of the major developments in the information systems areas. Relational data structures, generally implemented in Relational Database Management Systems (RDBMS), is based upon the relational algebra developed by Codd and San Jose (1972). The major advantage of using a relational data structure is its powerful query language for searching mechanisms. The use of relational data structures in workflow implementation is exemplified by Mangan and Sadiq (2002b).

While relational data structures are suitable in capturing bidirectional relationships (Whitten et al., 2007b), its performance in searching significantly drops when relationships are complex, as in the case of among process elements (Mangan et al., 2002b). For this reason, the use of relational data structures for implementation of linear expressions of processes for analytical purposes is an unsuitable approach for this research.

The other suitable candidate is the binary tree structures. Binary trees are considered suitable over generic tree structures, as there are well-tested analytical algorithms for binary tree manipulation (Knuth, 1973). Prior to relational databases binary tree structures were considered the most suitable data structure for representing complex relationships. In a book on data structures by Shave (1975) the mechanism to convert any generic tree structure into a binary tree structure is shown. This gives the advantage of using powerful and well-tested algorithms for initial creation of the tree and for adding and reading elements to and from tree structures. Therefore, the representation of linear expressions can be done using binary tree structures (Shave, 1975), in a manner that it can be analysed effectively as required by this research.
This section discussed a set of candidate MTTs that can be used for providing a solution for impact propagation analysis. In particular, this discussion evaluated strengths and weaknesses of a number of techniques and tools, in capturing the associations of process elements.

### 3.6 Chapter Summary

This chapter sets the foundation for this research among the substantial number of related research works in the workflow area. The specific area that needs further investigation or the ‘gap’ is identified to be, ‘**propagating impact analysis of process evolutions and changes, in web-based workflow systems**’.

This chapter initially sets the background by introducing a number of key terminologies and their usage in this research. Secondly, it introduces the theoretical framework that shows the context of the related research. Based on this theoretical framework, literature in a number of key related areas was analysed in detail. This analysis started with a general exploration of workflow technologies and process conceptualisation mechanisms. Next, work that supports process evolution was studied to find the state of the art developments in the area. The study is then extended to advanced topics in relation to workflow evolution such as workflow analysis and propagating impact analysis.

The findings reveal that there is significantly low number of research studies in relation to propagating impact analysis in process evolution. In addition, even the research that attempt to address this aspect fails to provide a holistic solution.

The final part of this chapter gives an overview of the possible theoretical concepts that may be used in formulating a solution to the problem of propagating impact analysis. The foundation for the solution is based on comprehension of various associations that process elements have among themselves and with other external artefacts that defines their existence or automate them. In addition, the possible formalisms or data structure that could facilitate capturing the above-mentioned complex relationships were analysed. These theoretical concepts are applied in an appropriate context in the subsequent chapters of this thesis.
Prior to the detailed discussion on building associations of process elements in Chapters-5 and 6, the next chapter introduces the running example that will be used in subsequent chapters to exemplify the use of the theoretical concepts.
Chapter-4

“*We have to do the best we know how at the moment. If it doesn’t turn out right, we can modify it as we go along.*”

*Franklin D. Roosevelt (1882-1945)*

4 Exploratory Case Study- OCAS Project

4.1 Chapter Overview

Most literature uses simple or hypothetical examples to research challenges and solutions associated with Business Process Evolution Management (BPEM). In contrast to this approach, use of relatively large and real-life examples for the same purpose, gives a practical perspective towards dealing with the complex problems that arise in BPEM.

This chapter presents the Case Study used to comprehend the depth and breadth of the problem of *propagating impact analysis* in BPEM. This case study is based on a Business Process Automation (BPA) experience of *Courses Approval Processes* (UWS, 2003-2007b) of *University of Western Sydney (UWS) – Australia*.

The automation of Courses Approval Process of UWS, was carried out under the project named *Online Courses Approval System – OCAS* (OCAS, 2006). The OCAS project was started when an organisation wide re-structuring was taking place at UWS. Although the OCAS implementation is completed now, the organisational re-structuring is continuing at UWS, affecting all of its processes, including the ones that are automated in OCAS.

Chapter-1 previously introduced a framework named *paradigm of process automation- PoPA framework* (Figure 1.1 in page 5). The *PoPA framework* categorises different representations of a business process during its lifetime, into four levels. The first level is the *pragmatic level* in which a business process is represented in contextual artefacts such as federal and state laws, regulations,
organisational structures, policies, guidelines and goals (Ramesh et al., 2005). Second level is the semantic level, which denotes practiced processes of an organisation. The syntactic level is the third level, which indicates various conceptual models created in relation to a process, for the purposes of automation or re-engineering (Senge, 1994). The fourth level is the implementation level that embodies the implementation artefacts of the automated process. The case study presented in this chapter is discussed in relation to these four levels of the PoPA framework.

The extensive literature review carried out in Chapter-3 proved the lack of research that had a holistic approach towards BPEM. A holistic approach to BPEM needs to cover two aspects as follows:

- The ability to manage evolutions that instigate at any one of the four levels (pragmatic, semantic, syntactic, and implementation) of the PoPA framework (van der Aalst et al., 2000).
- The capacity to manage evolutionary aspects of all process dimensions—actions, participants and business object (van der Aalst et al., 1999).

In this perspective, this case study chapter also focuses on understanding the different process dimensions (actions, participants, and business object) associated with the Courses Approval Processes of UWS. In particular, attention is given to understanding the evolutionary nature of these process dimensions.

The case study presented in this chapter is used to complement the constructive research method used in this work. In Chapter-2, the constructive research approach (Kasanen et al., 1993) was identified as the core research strategy to be used. In constructive research, problem identification and validating the practical utility of the solution are two of the important steps. The OCAS case study is the primary source that facilitates comprehending the challenges in relation to BPEM in web-based workflow systems. In addition, certain sample process evolution scenarios taken from OCAS are used (later in Chapter-8) in verification of the solution provided for the identified problem.

This case study chapter is organised as follows. This chapter first presents the background of OCAS project, with a justification on the selection of OCAS as the running example of this research. Then based on a process automated in OCAS, the pragmatic, semantic, syntactic, and implementational representations of processes
are discussed. Finally, this chapter presents some of the evolutions that took place in Courses Approval Processes during and immediately after the OCAS implementation. These evolutions are real-life examples that indicate the amount of changes that could occur in processes during the turbulent-times of an organisation.

4.2 OCAS background

Higher education institutions are constantly under pressure to become more efficient and effective. Thus, these institutions attempt to adopt new management systems and processes. In other industry sectors, automation of repetitive long running business processes has prevailed since the 1970s (zur Muehlen, 2004a). However, the wave of business process re-engineering and automation caught up with tertiary institutions in the 1990’s (Birnbaum, 2000).

Administrative processes of higher education institutions are usually good candidates for automation and streamlining. This chapter presents an automation effort of a typical administrative process of UWS-Australia, named Courses Approval Processes (UWS, 2003-2007b).

In UWS terms, unit is a subject area that is taught during one semester, such as ‘Introduction to Workflows’. A course is a collection of units. Students enrol and graduate from courses, such as ‘Bachelor of IT’, by completing a combination of compulsory and recommended sets of units. UWS offer courses locally in Australia in six campuses in the Western Sydney region and in overseas locations, in collaboration with overseas tertiary educational institutes. The courses offered in Australia are referred to as ‘on-shore’ courses and the overseas offerings are referred to as ‘off-shore’ courses.

There are mainly six approval processes, under ‘courses and units approval processes’. In the Courses Approval Policy (UWS, 2003-2007b) these processes are identified as follows:

(a) New Course Approval Process,
(b) New Unit Approval Process,
(c) Course Variation Approval Process,
(d) Unit Variation Approval Process,
(e) Course Discontinuation Process, and
(f) Unit Discontinuation Process.
The new course and unit approval processes (above (a) and (b)) ensure the newly introduced courses and units are financially viable, academically sound, and in line with the goals of the university. In order to meet various needs such as, new market trends, legal requirements, and competitor pressures; UWS recurrently modifies existing courses and units. Hence, course and unit variation processes (afore listed (c) and (d)) are there to make necessary changes in the best interest of the organisation and for the students. The introduction of new courses and units leads to retirement and withdrawal of existing ones. The course and unit discontinuation processes ((e) and (f) listed above) assure this retirement is a smooth transition for the students enrolled in those withdrawn courses and units.

These processes are usually considered as ‘manufacturing processes’ of tertiary education institutes. In these manufacturing processes, main ‘product’ – courses, are designed and created, to be offered to its ‘customers’ – students. In addition, from time to time that product is changed to meet the market needs. Over a period, certain products become obsolete and withdrawn from being offered to customers.

In mid 2005, the Academic Senate of UWS initiated a project to automate Courses Approval Processes. After a failed attempt to find an appropriate ‘off-the-shelf’ product for automation, the Academic Senate of UWS decided to endeavour this as an in-house development. Then, the task of automation was entrusted to Advanced enterprise Information Systems Management - AeIMS (AeIMS, 2005) research group, within the School of Computing and Mathematics (SCM) of UWS. The project was named OCAS - Online Courses Approval System (OCAS, 2006).

The main objectives of the OCAS project was to streamline and automate processes associated with Courses Approval Policy of UWS (UWS, 2003-2007b). With this objective, an extensive re-engineering effort was carried out prior to the automation, involving a number of participants that are managing and using the process. In this re-engineering phase, processes outlined in the policy documents were compared against the practiced processes in UWS. This investigation revealed significant differences between the two. Another important aspect identified was the evolutionary nature of the process. The evolutionary nature and lack of comprehension of the process steps by the participants were the main challenges that AeIMS research group had to address by an automated web-based workflow system.
In order to gain an insight to the challenges, AeIMS research group also studied some similar projects completed in other Australian educational institutions. One project that is similar to OCAS was a project accomplished in University of Technology Sydney (UTS) named as Online Courses Approval Project (OCAP, 2006). Similar to OCAS project, OCAP at UTS was aimed at automating the Courses Approval Processes. Although the OCAP implementation was completed and system was in operation, there was flexibility issues associated with it. Therefore, AeIMS research group had to find appropriate solutions to make this highly customised system to be adequately flexible to cope with the continuous changes that take place in the organisation.

AeIMS research group started the OCAS project in September 2005 and completed by December 2006. During this fifteen-month period, four processes in the Courses Approval Policy (new course and unit approval processes, and course and unit variation processes) were automated as the first phase. At the time of the writing this thesis, the Academic Senate of UWS is planning the wider rollout of OCAS in UWS.

Even before OCAS’s wider usage, it has been recognised as a successful project. In 2006, in the quality audit carried out by the Australian Universities Quality Agency (AUQA)\(^8\), highly commended OCAS project in their audit report as follows; “… AUQA commends UWS for the development of computer-supported quality systems for consolidating data and tracking processes including the Online Course Approval System (OCAS)…”(AUQA, 2007, p19-20).

The author of this thesis was a core member of the development team (of the AeIMS research group) which carried out the OCAS project. As the main business analyst of the team, the author was involved from the initial investigation to the final implementation of the project. In addition, the author was also involved in contributing to certain design tasks and supporting the testing activities in the latter stages of the project.

\(^8\) “AUQA is funded by the Commonwealth, State and Territory Governments of Australia with a vision to provide public assurance of the quality of Australia’s universities and other institutions of higher education” (AUQA, n.d)
4.2.1 Why OCAS as the Case Study?

There are three reasons that justify the use of OCAS as the case study in this research. These are:

- Complex and extensive nature of the business processes that were automated in OCAS.
- Evolutionary nature of the processes automated in OCAS.
- OCAS being implemented as a web-based workflow system.

Each of the above aspects is discussed in the following sections.

4.2.1.1 Complex and Extensive Business Processes in OCAS

As mentioned before, the use of simple and hypothetical examples, fails to address complex issues that arise in relation to managing evolutions in large processes. For example, when processes are simple (with about a dozen actions) it is relatively easy for business analysts to comprehend the propagating impact of a change. However, when processes are larger the same task is not straightforward. For this reason, this research required a case study that would facilitate exploring the nature of evolution of large-scale processes.

When analysing the three main process dimensions (actions, participants and business object) of the business processes automated in OCAS, it is evident that these processes are relatively large and complex. Following is an analysis of process dimensions of OCAS processes:

- **Actions** – On average, there are over 40 different actions associated with each process automated in OCAS. Human participants perform most of the semi-automated actions. In addition, there are automated tasks, such as sending e-mail, or triggering another process, which does not require human involvement. Further, there are up to five sub processes for each of the above listed processes. These processes are constructed using all possible main flow patterns, identified by van der Aalst et al. (2003b) - sequence, parallel split, synchronisation, exclusive choice, simple merge, multi choice, synchronising merge and multi merge.

- **Participants** – Organisational roles identify the process participants associated with each process (Russell et al., 2005b). In OCAS, there are about 50 different roles associated with the participants. These participants mainly belong to two groups; i) individual participants such as Head of the School,
Dean, Course Officer and Pro-Vice Chancellor and ii) groups of participants identified as College Education Committee members, External Advisory Committee members and Academic Senate members. These participants are identified in the two organisational structures of UWS; management structure (UWS, 2004b) and governance structure (UWS, 2004a).

- **Business Objects** – Course and Unit are the two business objects associated with OCAS processes. Each of these objects constitutes more than 200 attributes, which defines a course or unit. Some of the attributes are course name, unit name, course code, learning outcomes of the unit, and teaching staff. In each of the process actions, a subset of these object element instances will be created, edited, viewed, or modified, by a process participant.

Above statistics show that OCAS related processes are large in comparison to simple hypothetical examples.

### 4.2.1.2 Evolutionary and Changing Nature

During the fifteen-months of OCAS development and immediately after, the processes automated in OCAS continued to evolve and change, as UWS was going through an organisational re-structuring process. In order to accommodate strategic and operational decisions that were made from time to time, OCAS related processes had to change and evolve.

Evolution and changes in processes ultimately result in affecting the three process dimensions – actions, participants and objects. The changes in actions, participants and business objects, needs to be accurately reflected in the underlying workflow systems. This aspect of reflecting of process changes in automated systems, being the key question of this research, the *evolutionary nature* of courses and units approval processes also makes OCAS a suitable candidate to be used as the case study.

### 4.2.1.3 Automated as a Web-based Flexible Workflow

UWS is geographically dispersed with six campuses across the Western Sydney region of Australia (UWS-Campuses, 2004). The process participants of OCAS related processes are in all of these campuses. Prior to OCAS, the participants heavily used e-mail for messaging and communicating purposes of the processes. The ad-hoc nature of e-mail and lack of central coordination of activities created a
number of issues such as losing track of version controls of documents, actions not being carried out in the proper order or simply omission of actions. Therefore, it was required to choose a technology that not only could provide centralised coordination of activities, but also facilitate the participants across six campuses to engage in performing process actions effectively.

Due to the above-mentioned reason of dispersed participants and other reasons such as cost effectiveness, web-based technologies were considered as the obvious choice. In addition, AeIMS being a research group actively engaged in small to medium industry sector in Western Sydney region, already had a number of web technological frameworks that could fast track the project.

In Chapter-1, the scope of this research is identified to be focused on business processes that are automated as web-based workflow systems. In this perspective, OCAS being a web-based workflow system is the third reason that justifies its use as the case study of this research.

Next section discusses the OCAS implementation experience in relation to four levels – pragmatic, semantic, syntactic, and implementation, in the PoPA framework.

### 4.3 OCAS in the PoPA framework

As mentioned before, in the first implementation phase of OCAS, four processes (out of six processes covered in the Courses Approval Policy in UWS) were automated. These are *New Course Approval Process, New Unit Approval Process, Course Variation Process* and *Unit Variation Process*. These processes have strong similarities, from the implementation and evolution point of view. Therefore, to avoid repetition only one process (*New Course Approval Process*) is explained in relation to the four levels of information representations- pragmatics, semantics, syntactic, and implementation, of the PoPA framework.

#### 4.3.1 Pragmatics of the New Course Approval Process

UWS is a university in the Western Sydney region of Australia, it is compelled to be in accordance with related laws, and regulations set by both the New South Wales state government and Australian federal government. In addition, there
are a number of policies and regulations set within UWS. In the pragmatic level, these laws, regulations, policies, and guidelines are termed as contextual information. Therefore, there are a number of contextual artefacts that are applicable to the New Course Approval Process in OCAS.

Table 4.1 presents the external and internal contextual artefacts that define the New Course Approval Process in UWS. This table has three columns to capture; a) the name of the contextual artefact, b) name of the body (organisation, committee, ministry, or department) that is responsible for maintaining each of the context information and c) the internal or external nature of each contextual information.

<table>
<thead>
<tr>
<th>NAME OF THE CONTEXTUAL INFORMATION ARTEFACT</th>
<th>GOVERNING BODY OR POLICY APPROVAL AUTHORITY</th>
<th>EXTERNAL/INTERNAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>iii) UWS Strategic Plan and Vision</td>
<td>Board of Trustees of UWS</td>
<td>Internal</td>
</tr>
<tr>
<td>iv) UWS Courses Approvals Policy</td>
<td>Academic Senate of UWS</td>
<td>Internal</td>
</tr>
<tr>
<td>v) UWS Governance Structure</td>
<td>Board of Trustees of UWS</td>
<td>Internal</td>
</tr>
<tr>
<td>vi) UWS Management Structure</td>
<td>Board of Trustees of UWS and Senior Management of UWS</td>
<td>Internal</td>
</tr>
<tr>
<td>vii) Courses and Unit Approval Timeline Document pursuant to the Policy</td>
<td>Academic Senate of UWS</td>
<td>Internal</td>
</tr>
<tr>
<td>viii) Course and Unit Templates and Guidelines pursuant to the Policy</td>
<td>Academic Senate of UWS</td>
<td>Internal</td>
</tr>
<tr>
<td>ix) Course External Advisory Committee Policy</td>
<td>Academic Senate of UWS</td>
<td>Internal</td>
</tr>
<tr>
<td>x) Records Management Policy</td>
<td>Vice Chancellor of UWS</td>
<td>Internal</td>
</tr>
</tbody>
</table>

Table 4.1 identifies ten contextual information artefacts of which two are external and eight are internal to UWS. The two external elements are Australian Federal Governments Higher Education Support Act 2003 and Higher Education Provider Guidelines - pursuant to the Act, managed by the Department of Education, Science and Training (DEST, 2007) of the federal government of Australia. The Academic Senate of UWS manages the following four contextual artefacts out of the eight internal ones.
• UWS Courses approvals Policy,
• Courses and Unit Approval Timeline Document pursuant to the Policy,
• Course and Unit Templates and Guidelines pursuant to the Policy and
• Course External Advisory Committee Policy

The *New Course Approval Process* is shaped predominantly based on the above four contextual artefacts.

In addition, the other contextual information such as UWS Strategic Plan and Vision, UWS Governance Structure, UWS Management Structure and Records Management Policy also plays a supportive role in characterising the *new courses approval process*.

### 4.3.2 Semantics of the New Course Approval Process

The *semantic level* represents the process in practice or the *operational process*; defined according to the contextual artefacts in the *pragmatic level*. The participants refer to the same contextual artefacts for understanding their tasks in the process. However, based on our findings in OCAS and the literature (Lewis et al., 2007); it was evident that different participants had different perceptions towards the same process from the operations point of view. This was mainly due to reasons such as participants having different levels of experience or practices within their department, in relation to the *New Course Approval Process*.

Here the commonly practised, *New Course Approval Process* is presented to provide an overview. The *New Course Approval Process* has two major phases as; Course Notification of Intension (CNI) and Full Course Proposal (FCP).

In the CNI or the first phase:

- initially the Project Manager or Proposer prepares a CNI document
- Then Head of School (HOS) submits this CNI at the school level.
- Followed by HOS, Dean recommends the CNI, at the College level.
- The Vice Chancellor’s Advisory Committee (VCAC) gives the final approval for CNI in the first phase.
After the completion of the first phase, FCP or second phase proceeds as follows:

- Proposer creates a FCP document and submits to the HOS
- HOS submits the FCP at the school level
- College Education, Assessment and Progression Committee (CEAPC) assesses the FCP at the college level
- Dean recommends the FCP at the college level
- Courses Approvals and Articulation Committee (CAAC) endorse the FCP
- Academic Senates gives the final approval on the FCP

In FCP phase when it is submitted to the HOS the following sub processes are initiated:

- **Resources Assessment (RA) Approval**: To ensure the required resources to offer the course are available
- **Business Case (BC) Approval**: To assess the financial feasibility
- **External Advisory Committee (EAC) Approval**: To gauge the suitability of the proposed course against the market needs
- **Internal Forum (IF) Process**: To certify that the proposed course does not clash with existing courses from academic and objectives point of view.

This two-phased *New Course Approval Process* is illustrated in Figure 4.1.
Figure 4.1- High-level New Course Approval Process of OCAS- adopted from AUQA Report (AUQA, 2007)
Each of the above-mentioned process actions has specific reasons for being present in the New Course Approval Process. Broadly, these reasons are categorised into three: value adding, gate keeping and communication.

- **Value adding actions** contributes to achieving the end goal of the process, which is creating an academically sound competitive course that helps shaping the careers of students and profitable for UWS.

- **Gate keeping actions** perform certain checks and balances before the process moves across organisational units, such as from school level to college level.

- **Communicative actions** predominantly inform interested parities, who are not actively involved in the process or trigger other processes.

In some cases, certain actions can be present in the process to fulfil more than one of the above-mentioned reasons. For example, the action of HOS’s submission of CNI not only adds value to a course, but also performs the gate-keeping function at the school level.

When processes are introduced based on contextual information, it fails to capture rationales behind each process steps, discussed-above. This inability to capture the rationale behind process actions leads to the first leakage of information in the transition from pragmatic level to the semantic level, as it was discussed under PoPA framework previously in Chapter-1.

Table 4.2 summarises the leakage of information in relation to the actions of the New Course Approval Process.

<table>
<thead>
<tr>
<th>ACTION</th>
<th>PARTICIPANT</th>
<th>LEAKING INFORMATION</th>
<th>RATIONALE / TYPE OF ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create CNI</td>
<td>Project Manager</td>
<td>Create a CNI document</td>
<td>Value adding</td>
</tr>
<tr>
<td>Submit CNI</td>
<td>HOS</td>
<td>Submits the CNI to the Dean</td>
<td>Value adding and Gate-Keeping (at school level)</td>
</tr>
<tr>
<td>Recommends CNI</td>
<td>Dean</td>
<td>Recommends the CNI at the college level</td>
<td>Gate-Keeping (at college level)</td>
</tr>
<tr>
<td>Approve CNI</td>
<td>VCAC</td>
<td>Final Approval on the CNI after assessing the financial viability of the course</td>
<td>Value adding and Gate-Keeping</td>
</tr>
<tr>
<td>Create FCP</td>
<td>Project Manager</td>
<td>Create the FCP document and related other course</td>
<td>Value adding</td>
</tr>
<tr>
<td>ACTION</td>
<td>PARTICIPANT</td>
<td>DESCRIPTION</td>
<td>RATIONALE / TYPE OF ACTION</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>Submit FCP</td>
<td>HOS</td>
<td>Submits the FCP to the CEAPC</td>
<td>Value adding and Gate-Keeping (at school level)</td>
</tr>
<tr>
<td>Assess FCP</td>
<td>CEAPC</td>
<td>Assess the academic soundness of the proposed course and its alignment with the academic goals of the college</td>
<td>Value Adding</td>
</tr>
<tr>
<td>College Recommend FCP</td>
<td>Dean</td>
<td>Recommends the FCP to at the college level</td>
<td>Gate-Keeping (at college level)</td>
</tr>
<tr>
<td>Recommend on FCP</td>
<td>CAAC</td>
<td>Checks the courses alignment with the academic goals of the university</td>
<td>Value adding</td>
</tr>
<tr>
<td>Final Approval on FCP</td>
<td>Academic Senate</td>
<td>The final approval based on the recommendations made by previous process participants</td>
<td>Gate-Keeping (at university level)</td>
</tr>
</tbody>
</table>

The later chapters of this thesis will discuss the issues that arise from the above-mentioned information leakage identified in Table 4.2.

4.3.3 **Syntactic Models of the New Course Approval Process**

In general, there are three types of models created in system development. These are: *process models, business object or data models*, and *interface definitions* (Whitten & Bentley, 2007a). The *syntactic level* of information representation represents these models created in system development.

Usually there are industry standard tools advocated to be used for modelling purposes, such as UML models (OMG, August 2005). However, in some occasions the use of industry standard models are not advisable due to practical reasons, such as stakeholders’ and process managers’ familiarity on those tools. Therefore, in OCAS, most modelling tools were customised to suite the needs of the stakeholders of the project.

Below these syntactic models used in OCAS are discussed. In this discussion, particular attention is given to demonstrate how certain tools were tailored to suite the needs of OCAS and its stakeholders.
4.3.3.1 Process Models

The methodology used in OCAS development distinguishes two steps associated with process modelling (Ginige, Murugesan, & Kazanis, 2001). The first step concentrates on capturing the process as-it-is into abstract models. The second step focuses on re-engineering the process, with the participation of process managers and end-users. For the successful completion of these steps, it is vital to use a tool that is suitable for the modelling task and simple enough to be understood by process managers, who are not information system experts.

Initially the author (as the business analyst of the OCAS project) assessed a set of industry standard high-level process modelling tools such as UML (Unified Modelling Language) activity diagrams (OMG, August 2005) and Petri-Nets (van der Aalst, 1997; Murata, 1989). After this analysis, the development team decided to use UML activity diagrams, which they thought were simple for the process managers and end-users to understand. However, stakeholders, process managers, and end-users who were experts in areas such as nursing, education, psychology, law, and history other than Information Systems, found it difficult to comprehend UML activity diagrams. This was evident when summarising the feedback received from the first four meetings conducted using around 20 participants. About 40% feedback suggested improving the process diagrams used and only 10% were satisfied with the UML activity diagrams. About 50% had not commented on the diagrams. Further, some users mentioned their comfort with more text combined with less use of complex notations.

As a result, the business analysts identified some other modelling tools, which suited both technical requirements and high-level business requirements. These findings led to the creation of two OCAS specific tools to represent high-level and detailed process models. The high-level process-modelling tool was a simplified version of Event-driven Process Chains (Mendling et al., 2005a). The technical process definition tool was a tabular format similar to the State Tables (Mano, 1988), which were originally created based on the theory of finite state machine by Mealey (1955) and Moore (1956).

This section, initially details the high-level process modelling tools used in OCAS project.
The high-level process models used in OCAS are a slightly modified version of Event-driven Process Chain (EPC). These models visually capture the state of the models using a circle notation (see Figure 4.2). The other information related to this state, such as actions, associated business object and process participants who can perform actions are linked and presented in textual format.

![Figure 4.2- Simplified EPC diagrams used in OCAS high-level Models](image)

The process managers and end-user of OCAS referred to these diagrams as ‘bubble charts’ and most importantly were able to comprehend the process using these diagrams. Figure 4.3 gives the full set of notations used in these simple process flow diagrams. In Figure 4.3, the labels from A to I show following flow constructs.

- **A** - Double lined circles denote the *start state* of a process. Inside the state, the action, which leads to the next state, is given in text format.
- **B** - Single lined circles denote the intermediate process steps.
- **C** - Thick boarded circles denote the end state of a process.
- **D** - Solid lines that connect two different states denote the natural flow actions from one state to another.
- **E** - Dotted lines that connect two different states denote the merging of two or more sub-processes.
- **F** - Solid line that points to the same state shows the iteration of a task.
- **G** – Two parallel lines start from one state and ending in two different states denote the parallel split.
Figure 4.3- Description of Notations used for modelling high-level processes in OCAS

- **H** - Two parallel lines start from a diamond shape and ending in two different states show the conditional split of events.
- **I** – Solid lines starting in different states ending at one state denote the merging of events.

Figure 4.4 illustrates the high-level process view of the *New Course Approval Process* using the above-listed diagrammatic notations. In this diagram, labels 1-32 are used for explanatory purposes and will be later referenced.
Figure 4.4 - High-Level Process Diagram for the New Course Approval Process
The process diagram in Figure 4.4 of the *New Course Approval Process* depicts the forward process path. It also denotes the process participants. Table 4.3 lists the full set of actions and the roles of the associated participants who perform these actions, in a much more readable format. The numbers given in column one corresponds to the labels used in Figure 4.4.

**Table 4.3- Full set of Actions and Participants of the *New Course Approval Process***

<table>
<thead>
<tr>
<th>NO AS IN Figu re 4.4</th>
<th>ACTIONS NAME</th>
<th>ROLE OF THE PARTICIPANT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Create Course Notification of Intention (CNI)</td>
<td>Proposer</td>
</tr>
<tr>
<td>2.</td>
<td>Submit CNI</td>
<td>Head of School</td>
</tr>
<tr>
<td>3.</td>
<td>Recommend CNI</td>
<td>Dean</td>
</tr>
<tr>
<td>4.</td>
<td>Approve CNI</td>
<td>Pro vice Chancellor (Academic)</td>
</tr>
<tr>
<td>5.</td>
<td>Create Full Course Proposal (FCP)</td>
<td>Proposer</td>
</tr>
<tr>
<td>6.</td>
<td>Create unit information</td>
<td>Academics</td>
</tr>
<tr>
<td>7.</td>
<td>Submit FCP</td>
<td>Head of School</td>
</tr>
<tr>
<td>8.</td>
<td>Assess academic quality and recommend FCP</td>
<td>Chair of College Education, Academic and Progression Committee (CEAPC)</td>
</tr>
<tr>
<td>9.</td>
<td>College Recommendation for FCP</td>
<td>Dean</td>
</tr>
<tr>
<td>10.</td>
<td>Course Approvals Committee Recommendation for FCP</td>
<td>Chair of Courses Approvals and Articulation Committee (CAAC)</td>
</tr>
<tr>
<td>11.</td>
<td>Approve FCP</td>
<td>Chair of Academic Senate</td>
</tr>
<tr>
<td>12.</td>
<td>Fill in the course code information</td>
<td>Course Officer</td>
</tr>
<tr>
<td>13.</td>
<td>Commence internal forum consultation</td>
<td>Proposer</td>
</tr>
<tr>
<td>14.</td>
<td>Submit internal forum consultation report</td>
<td>Associate Dean (Academic)</td>
</tr>
<tr>
<td>15.</td>
<td>Commence external advisory committee consultation</td>
<td>Proposer</td>
</tr>
<tr>
<td>16.</td>
<td>Submit external advisory committee consultation report</td>
<td>Associate Dean (Academic)</td>
</tr>
<tr>
<td>17.</td>
<td>Create and submit Resource Assessment (RA) documentation</td>
<td>Proposer</td>
</tr>
<tr>
<td>18.</td>
<td>Office of Marketing Sign-off on the RA document</td>
<td>Director of Marketing</td>
</tr>
<tr>
<td>19.</td>
<td>IT directorate’s (ITD) sign-off on the RA document</td>
<td>Director of ITD</td>
</tr>
<tr>
<td>20.</td>
<td>Education Development Unit’s (EDU) sign-off on the RA document</td>
<td>Director of EDU</td>
</tr>
<tr>
<td>21.</td>
<td>Library sign-off on the RA document</td>
<td>Information Systems Librarian</td>
</tr>
<tr>
<td>22.</td>
<td>Office of Academic Registrar – System’s (OAR-Sys) sign-off on the RA document</td>
<td>Assistant Academic registrar systems</td>
</tr>
<tr>
<td>23.</td>
<td>Office of Academic Registrar – Operation’s (OAR-opr) sign-off on the RA document</td>
<td>Assistant Academic registrar operations</td>
</tr>
<tr>
<td>24.</td>
<td>Timetabling sign-off on the RA document</td>
<td>College timetabling officer</td>
</tr>
<tr>
<td>25.</td>
<td>Capital Works sign-off on the RA document</td>
<td>Senior planning architect</td>
</tr>
<tr>
<td>26.</td>
<td>Approve RA document</td>
<td>Dean</td>
</tr>
<tr>
<td>27.</td>
<td>Create Business Case (BC) document</td>
<td>Proposer</td>
</tr>
<tr>
<td>NO AS IN FIGURE 4.4</td>
<td>ACTIONS NAME</td>
<td>ROLE OF THE PARTICIPANT</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td>28.</td>
<td>Assess and Sign-off the BC with an independent statement</td>
<td>Business Development Officer</td>
</tr>
<tr>
<td>29.</td>
<td>Recommend BC</td>
<td>Dean</td>
</tr>
<tr>
<td>30.</td>
<td>Vice Chancellor’s Advisory Committee Approval for BC</td>
<td>Pro-Vice Chancellor (Academic)</td>
</tr>
<tr>
<td>31.</td>
<td>Business and Finance Committee Approval for BC</td>
<td>Deputy-vice chancellor (development and international)</td>
</tr>
<tr>
<td>32.</td>
<td>College Approval for BC</td>
<td>Dean</td>
</tr>
<tr>
<td>33.</td>
<td>View course details</td>
<td>UWS community (which include all personal who has got a staff number in UWS)</td>
</tr>
</tbody>
</table>

The high-level process models created (as in Figure 4.4) were extensively used for re-engineering and many other communication purposes, between the stakeholders and the development team. However, the details provided in these ‘bubble charts’ were not sufficient for process managers, who were involved in the day-to-day operations of the process. In particular, showing how the process behaves when one makes a decision that moves the process in alternative directions, such as rejecting or asking for modifications in a course proposal. The following example shows the alternative paths that a process can take from a particular state.

When Dean performs the action recommend CNI he/she can make three decisions. These are to either ‘recommend CNI’ or ‘oppose the CNI’ or ‘request for a re-submission of CNI’. The diagram in Figure 4.4 only illustrates the behaviour based on the decision ‘recommend CNI’. The decision to ‘oppose the CNI’ creates a sudden death of the process and the choice to ‘request for a re-submission’ sends the process a step back. However, Figure 4.4 does not depict these alternative paths that are vital for the process administrators and developers. The author (as the business analyst) in consultation with developers, created a tool to capture additional information. This tool in tabular format, records states of high-level process diagrams, forward path, and alternate paths.

This concept of tabular representation of process was inspired and adopted from automata state machine theories by Mealy (1955) and Moore (1956) and used as State Tables by Mano (1988). The process managers and developers of OCAS used to refer to this tabular format of process representations as ‘State Tables’ (see Figure 4.5).
Figure 4.5- 'State Table' representation of Process Details
In a State Table, the first row shows the column headings and there are ten (10) columns. These columns capture five types of information as follows:

- Details on current state of the process,
- Details on the actions performed at the entry to the state,
- Details on the actions performed while in state,
- Details on actions performed at the exit from the state and
- Details on next state of the process

A null entry in a cell indicates that the content in that cell is not applicable and an empty cell denotes the inheritance of value from the upper cell. The ten columns used in the State Table are described below.

- **Current State**: This is the very first column in the State Table, which records the states associated with the process. Entries in the *current state* column have a direct correlation to each of the circles in the high-level process in Figure 4.4. For each bubble in Figure 4.4, there is a corresponding current state entry in this column. In addition, there are extra states not explicitly identified in the high-level diagram. These extra states are reject, end, and intermediate (where the processes are waiting for merging to take place). All these process states are represented in the current state column, with a unique name. There is a specific format for the state name as ‘unique number--descriptive state name’. The unique number is just a short label for the state, which allows easy reference to the state in other places of the State Table. The descriptive state name is a label that has a meaningful name in relation to the process. For example, a state name such as ‘3--notification waiting for HOS decision’ has the unique number 3, and textual description which informs the reader the business context of the state.

- **Entry Condition**: The second column is the *entry condition*. There are two uses for this entry condition. Firstly, it restricts the execution of automated actions on entry in the third column. Secondly, it also allows to set pre-conditions for allocating instate actions in the fifth column, to the process participants in the fourth column. For example, when entry condition is onshore=1 (that is when courses are offered locally in Australian campuses) the Dean is allowed to perform the instate action of Dean’s executive veto for course notification. This also means that when a course is proposed to be offered at overseas locations
(offshore courses) the *Dean* is not allowed to use his/her vetoing privileges. This column can have compound conditions using normal logic operators such as and, or and xor.

- **Automated Actions on Entry**: The name of the third column is *automated actions on entry*. Any action performed at the entry to the state, such as sending e-mails and/or triggering another process, is in this column. If there are more than one action they are separated by asterisks (*).

- **Actor**: The fourth column named *actor*, captures the organisational roles, of the participant. These participants can perform the instate actions in the next column. There can be multiple participants allowed to perform the same action. When multiple participants perform the same action, it requires repeating the rows for each actor as shown in Figure 4.6. In this example, the action *Head of School approval for resource assessment* can be performed either by the *Head of School* or by the *Associate Head of School-Academic*.

![Figure 4.6- Same Action Performed by Multiple Participants](image)

- **Actions**: The fifth column indicates the actions that require human intervention. For each action in this column, there is a corresponding action interface. OCAS being a web-based system, these action interfaces are web-forms, which allow the participant to interact with the system. Depending on the state of the process and role of the participant, each of these web-forms will give granular level access to
the business object of the process. This means that only a set of fields can be edited or viewed and other fields are hidden from the user.

- **Form Events**: This sixth column captures the *form events or the buttons* associated with the web-form presented in the fifth column. Every web-form will have form events such as *submit* and *cancel*. In OCAS these form events represents the decisions that process participants need to make. For example, *Head of School* might decide the request for a re-submission of CNI after reviewing it. In that case, the HOS can press the *re-submit* button. Similarly, there are *reject* and *approve* buttons, that allows the participant to either reject course documentation or to approve it. The buttons that needs to be shown at the end of the form can be dynamically generated based on the entries made in this column, at run-time.

- **Exit Condition**: This column captures the conditions that need to be satisfied before allowing the process to exit the current state. These conditions could be similar to checking whether another process has reached a particular state or value of a variable.

- **Actions on Exit**: The column *actions on exit* capture certain automated actions, such as sending e-mail or activating a web service, while transiting to another state.

- **Transitional Conditions**: This column sets an additional set of conditions that is to be checked before transiting from one state to another.

- **Next State**: The final column is the *next state* column. This column indicates where this process will next transit, depending on the actor who performed the action, form-event or the button pressed and transition conditions. In this column, the short label given in the first column denotes the next state.

The State Tables can handle the common control flow patterns such as sequential, conditional split, parallelism and merge.

The sequential and conditional flow of actions is handled using the columns-current state, actor, action, form event, exit condition, and next state. Figure 4.7 shows a portion of a State Table, used to explain sequential and conditional flow.
According to Figure 4.7, ‘project manager’ performs the action ‘update course notification’, in the state named ‘2-initial’. The action ‘update course notification’ has two form events namely save and submit. When the project manager presses the submit button on the form, the process moves to the next state, which is named as ‘3-notification waiting HOS decision’. This shows the sequential flow of actions from ‘update course notification’ to ‘HOS approval for course notification’.

In the new state ‘3-notification waiting HOS decision’, the HOS performs the action ‘HOS approval for course notification’. Here there are three buttons - approve, reject or re-submit that the HOS can press. Upon the pressing of approve button, the process has a conditional flow. This conditional flow is guarded by the exit condition that checks the value of the course attribute offshore (a course is proposed to be offered in overseas locations). If Offshore=1, the process moves to the next state labelled in short as 4 and if Offshore=0 to the state labelled as 5.

To choreograph the parallel split into sub processes and merge, multiple State Tables are used. For example, in the resource assessment process there are eight sign-offs required by various resources owners. These sign-offs associated in this process needs to take place in parallel. Hence there are smaller State Tables for each of these parallel process paths as shown in (d) in Figure 4.8. The (b) shows the state of the resource assessment process where it waits for the parallel processes to complete. The exit condition that checks the state of the parallel processes is denoted by (c) in Figure 4.8.
The above-discussed State Tables capture the processes in detail, in contrast to high-level bubble charts. These State Tables have strict syntactic rules in writing the process. However, similar to any other process-modelling tool, the creative use of these State Tables to model complex processes is with the business analysts.

Because of the strict syntax used in State Tables, these can be programmatically processed. A parser that is capable of understanding the syntax of State Tables can create machine-readable process definitions, using this tabular representation of the process. Machine-readable process definitions used in OCAS are discussed later in the next section.

4.3.3.2 Object Models

The data model or the business object model is one of the important models in any information system project. OCAS, uses standard object-modelling tool, UML class diagrams, for abstracting the business object associated with the process. Figure 4.9 illustrates the high-level course object model in OCAS. As the focus of this research is not on data object modelling the details in relation to the object model is kept to a minimum.
Smart Business Object – SBO tool (Liang & Ginige, 2006) supported the implementation of the object model. Details about the SBO tool and its modelling language will be covered in the next section.

Figure 4.9: High-level Object model of the ‘Course’ Business Object of the New Course Approval Process

4.3.3.3 Action Interface Models

In process-oriented applications, most user interfaces (UI) are essentially the action interfaces. For each of the actions identified in the process model, it is required to define UIs.

In web-based workflows, action interfaces can be implemented as static HTML web-forms, without having to first model them. However, such an approach limits evolution or flexibility of the process. To achieve higher flexibility, OCAS uses a simple UI modelling tool, which combines both process model and business object model. Previously discussed object model and the process models are matched together to create action interface definition models.

In the UI model, a) each process state, b) actions identified in that state and c) participant indicated to perform the action taken from the process model, are mapped against the full attribute list derived from the data object model (Figure 4.11).
In OCAS, eleven (11) notations were used to explain the visibility aspect of each business object attribute in process actions, as shown in the legend of Figure 4.10. This visibility aspect, defines the method (such as hidden, editable text box, static text box, selection box, radio button and in report format) in which the field can be shown in the action interface. In addition, if the action interface is split into several screens (split form), it is required to denote the screen in which the field element is shown. For example, as highlighted in Figure 4.10, the course type field is presented as a selection box in the first screen of the stepped web-form, for the action create course notification.

This section discussed the syntactic models created in automation of the New Course Approval Process in OCAS. The next section describes the implementation details of OCAS.

### 4.3.4 Implementation of OCAS Processes

Flexibility of the processes was the focus in OCAS implementation. In this view, AeIMS decided to use their own web-based application development and
deployment environment. This application development framework is named as CBEADS® (Component Based E Application Development/Deployment System) (Ginige & Silva, 2007; Ginige & X. Liang, 2007). Use of CBEADS® framework to achieve maximum flexibility in OCAS implementation, is discussed next.

### 4.3.4.1 CBEADS®

CBEADS® is a framework, that allows the end-users to manage their application with the help of a set of tools. In CBEADS®, designers and developers create a set of tools to develop applications and end-users or business analysts can use these tools to create the process-oriented application. This approach used in CBEADS® is shown in Figure 4.11.

![Figure 4.11- Application Development approach used in CBEADS®](image)

According to Figure 4.11, (A) indicates the traditional approach of going through the normal system development life cycle to create monolithic applications. Label (C) indicates the other extreme of having high-end development tools to develop web applications, which require expertise in using those tools. CBEADS® takes the middle path, in which simple tools are created by designers and developers and domain users utilise these tools to create their own applications. Therefore, it is possible for both developers and end-users to be developing and deploying applications at the same time in a single framework.
Once programmers develop the high-level tools these can be re-used in other application development purposes. This concept is exemplified in Figure 4.12, in which the form generation tool, routing engine, and report generation tool created for a leave processing system being re-used for a purchase requisition system. Figure 4.12 distinguishes the two levels as application level and meta-application or the middle tier where all the re-useable tools reside.

**Figure 4.12- Re-use of Tools for end-user Development**

4.3.4.2 OCAS Implementation Architecture using CBEADS©

Figure 4.13 illustrate the implementation architecture of OCAS, CBEADS© as the underlying framework. The middle-tier provides several high-level tools for building the OCAS application components and for the enactment of them. These modules capture the features that could be used by other applications similar to OCAS.

**Figure 4.13 - OCAS Implementation Architecture**
Following are the specific tools used in OCAS for application development and runtime operations.

- **Security Sub Module** – According to the rules defined in the State Tables certain participants are allowed to carry out certain tasks depending on the state of the process. The security sub module takes care of maintaining these permissions, which are called capabilities.

- **Role Management System (RMS)** – RMS takes care of maintaining the organisational structure related information. This separation of role management, allows organisational structures to evolve independent of the OCAS application.

- **Smart Business Object (SBO)** – SBO handler allows the high-level management of information objects associated with the courses approval process (Liang et al., 2006). The key feature of this SBO tool is its ability to change the underlining data structure and reflecting those changes in the appropriate interfaces, without having to touch any code.

- **Workflow Management System (WfMS)** – This module is similar to any workflow engine that enacts the process according to the rules defined in the process definition. This WfMS requires machine-readable process models for it to control the routing of OCAS processes. These machine-readable process models are generated using previously explained State Tables. The Excel State Tables are run through a parser that generates the machine-readable process in a format as shown in Figure 4.14. In the machine-readable format, each state in the State Table is referred to as a hook and uses the unique identification number used in the State Table. The actions that can be performed on entry, instate, exit and transition, are respectively placed in the entry_hook, instate_hook, exist_hook and transition_hook.
The implementation approach and the architecture presented above allow greater flexibility to change the processes automated in OCAS. For instance, when real-life processes change, the State Tables, Object models, and Interface models can be changed in OCAS, for the new process-instances to follow the changed process. However, two limitations are associated with the current implementation of OCAS.

- **Issue of managing the dynamic evolutions**: In the present implementation of OCAS, each process instance has a process definition that is bound to it at the start. However, some process instances might take between 3 and 18 months to complete. During this time, the real-life processes can change and evolve. The new process definitions will be introduced to the WfMS. Although new definitions are introduced, already started processes require completing using the
old definition, which was initially bound to it. The inability to handle the process instances according to the changing process models is the first limitation of the current OCAS implementation.

- **Issue of finding the propagating impact:** The second issue is the effectiveness of changing process models to reflect real-life process changes. The real-life process evolutions need to be reflected in appropriate models (process, data and interface), by a human (either a process manager familiar with the implementation or the business analyst). However, the complex nature of the processes makes it an arduous task for humans to do the changes without creating any semantic errors or inconsistencies. For example, deletion of a certain data attribute, from the business object, may have effects on actions or routing conditions that depend on the removed attribute values. This aspect of creating errors or inconsistencies, by changing models, in different models or different places of the same model, is termed as *propagating impact*. The initial implementation of OCAS does not have a mechanism to identify the propagating impact of a change.

The issue of managing dynamic evolutions in processes is a highly researched area that does not require to be further covered in this work. However, as revealed in the literature review chapter (Chapter-3), the issue of propagating impact analysis is sparsely researched. Motivated by this specific problem in OCAS and the lack of research in the area, this research is focused towards finding solutions for identifying propagating impact of business process evolutions.

This section presented the OCAS in relation to four levels (*pragmatic*, *semantic*, *syntactic*, and *implementation*) in the *PoPA framework*. The next section reveals the certain real-life evolution scenarios that took place in processes of OCAS, during its implementation and immediately after that.

### 4.4 Evolutions and Changes in OCAS Processes

Table 4.4 lists the changes and evolutions that took place in UWS that affected the OCAS related processes. These changes took place during and immediately after the implementation, which was close to two years. Table 4.4
explains the nature of the change, the level in the PoPA framework, in which the change was initiated and the immediately affected process dimension (action, participant, or object).

Table 4.4- Evolutions and Changes that affected the OCAS Processes

<table>
<thead>
<tr>
<th>Change or the Evolution</th>
<th>Description/ Reason</th>
<th>Initiated level</th>
<th>Immediately affected process dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previously identified role of course officer is now amalgamated into new role as</td>
<td>To include various other responsibilities associated with the role</td>
<td>Pragmatic level</td>
<td>Participants</td>
</tr>
<tr>
<td>Course Data Management Officer (CDMO)</td>
<td></td>
<td>level strategic</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>reasons</td>
<td></td>
</tr>
<tr>
<td>Adding attributes to gauge the courses viability in international markets.</td>
<td>Attributes are identified to check courses ability to be effectively offered outside Australia</td>
<td>Pragmatic level</td>
<td>Business Object</td>
</tr>
<tr>
<td></td>
<td></td>
<td>due to strategic</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>reasons</td>
<td></td>
</tr>
<tr>
<td>Delegation of financial approvals to executive Deans</td>
<td>Previously, the Deputy Vice Chancellor for developments centrally handled all the financial approvals in relation to a course. With decentralised budget management system, that gave Deans executive powers, it required Deans to decide on the final approvals only for fully domestic courses.</td>
<td>Pragmatic level</td>
<td>Process actions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>due to strategic</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>reasons</td>
<td></td>
</tr>
<tr>
<td>Introduction of a sub process for internal consultation purposes</td>
<td>In the previous process, the consultation among Academics, schools, and colleges were handled in an informal manner. This resulted in consultation process not taking place. Consequently, courses and units were introduced with repeating content. To avoid this, internal consultation process had to be formalised.</td>
<td>Semantic level</td>
<td>Process actions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>due to operational</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>reasons</td>
<td></td>
</tr>
<tr>
<td>Inclusion of Vice Chancellors Advisory Committee (VCAC) approval in the first phase of approval</td>
<td>In order to avoid Academics and other staff inputting time on courses that does not fit within the UWS strategic plans, it required some early inspections by higher management. This resulted in introducing a VCAC early in the planning stage.</td>
<td>Semantic level</td>
<td>Process actions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>due to operational</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>reasons</td>
<td></td>
</tr>
<tr>
<td>Introduction of College Education, Assessment and Progression Committee (CEAPC)</td>
<td>The colleges previously had a board of studies, in which college education committee was a sub committee in it. The re-shaped organisational structures introduced the CEAPC, which had more decision powers over courses</td>
<td>Pragmatic level</td>
<td>Participants</td>
</tr>
<tr>
<td></td>
<td></td>
<td>strategic reason</td>
<td>(CEAPC represents a group of participants)</td>
</tr>
<tr>
<td>Previously known as</td>
<td>To recognise the wide range of</td>
<td>Pragmatic level</td>
<td>Participants</td>
</tr>
<tr>
<td></td>
<td></td>
<td>reasons</td>
<td></td>
</tr>
<tr>
<td>Change or the Evolution</td>
<td>Description/ Reason</td>
<td>Initiated level</td>
<td>Immediately affected process dimension</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------</td>
<td>-----------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>Courses Approval Committee (CAC) is changed to Courses Approvals and Articulation Committee (CAAC)</td>
<td>activities carried out by this committee other than course approvals</td>
<td>level strategic reasons</td>
<td>(CAAC represents a group of participants)</td>
</tr>
<tr>
<td>Introduction of a new division named Student Administration – Services.</td>
<td>This amalgamated previously known Academic registrar systems and Academic registrar operations into a single unit</td>
<td>Pragmatic level strategic reasons</td>
<td>Participants</td>
</tr>
<tr>
<td>Inclusion of marketing division into the sign-off process</td>
<td>This is to get an understanding on the marketability of a course or to shape to suite the current market needs</td>
<td>Semantic level due to operational reasons</td>
<td>Actions</td>
</tr>
<tr>
<td>Precise costing methods to calculate the cost and revenues</td>
<td>More realistic costing methods were required to get an understanding of the financial viability of a course. Thus, new costing methods are introduced</td>
<td>Semantic level due to operational reasons</td>
<td>Business object</td>
</tr>
</tbody>
</table>

A sample of above listed evolutions will be selected for validating the solutions (later in Chapter-8) provided in this thesis.

### 4.5 Chapter Summary

This chapter discussed the main Case Study used in this research. This case study is primarily used as an exploratory case study for comprehending the depth and breadth of the problem of analysing the impact of process evolutions. The case study was based on an automation exercise carried within UWS, in streamlining their Courses Approval Processes, in a project named OCAS (Online Courses Approval System).

In this chapter, first the project background was introduced, with a justification of the use of OCAS as the case study. Followed by that, different types of process representations from pragmatic level down to implementation level were discussed. In this discussion, particular attention was given to illustrating certain modelling tools used to achieve maximum possible flexibility, of the automated processes. Further, this chapter also listed the real-life evolutions that took place in OCAS processes during the last two years.

Despite the adaptability nature in the implementation, lack of support for a mechanism that allows the analysis of propagating impact of process evolutions, was
identified as one of the issues associated with current OCAS implementation. In this view, the rest of the chapters of this thesis aim to provide a solution to the question “How can organisations, accurately and effectively reflect business process evolutions and changes in web-based workflows?”

The proposed solution is founded on the principle of totality, which refers to ‘nothing in this world exists in isolation’. In this view, the next chapter explores different kinds of associations that process elements have with contextual artefacts that define them, models that conceptualise, and workflow artefacts that implement them.
Chapter-5

“Relationships are all there is. Everything in the universe only exists because it is in relationship to everything else. Nothing exists in isolation.”
Margaret J. Wheatley

5 Constraints, Associations, and Dependencies (CAD) of Process Elements

5.1 Chapter Overview

In the eye of the critical philosopher, nothing in this world exists in isolation (Orlikowski et al., 1991). In other words, its associations with other elements that surround it define the very existence of an element. These relationships and dependencies play an important role in managing evolutions of any entity within information systems or business processes that are made up of correlated elements. In this view, this chapter discusses various Constraints, Associations, and Dependencies (CAD) that exists among business process elements.

A thorough comprehension of CAD of process elements, allows tracking the propagating impact that can gets created when one element is changed. For example, with information systems, changes in a database schema, such as removal of an attribute from a table, may result in having to alter the code files that interact with that table. Similarly, a change in an individual process dimension (actions, participants or business object) creates propagating impacts onto other dimensions. In-depth comprehension CAD of process elements allows locating such propagating impact, prior to changing already implemented workflow artefacts.

This chapter explores CAD of process elements, within the framework of paradigm of process automation- PoPA framework, previously introduced in Chapter-1 (Figure 1.1 in page 5). There are four levels of representations of a process
identified in the *paradigm of process automation*. These are; **pragmatic level** that denote contextual information elements (Laws, Policies and organisational structures), **semantic level** that indicate the operational business processes, **syntactic level** that represent the models created for automation and **implementation level** that symbolise implemented web-workflow artefacts.

The elements in each of the above four levels (**pragmatic, semantic, syntactic, and implementation**), have two types of associations; **Intra-level** and **Inter-level**.

The same level associations are referred to as **Intra-level CAD** in the *PoPA framework*. For instance, one organisational policy – ‘*recruiting of employees*’ which is a contextual element in the **pragmatic level**, has associations with another contextual artefacts - ‘*organisational structure*’. Such **Intra-level** associations exist within all four levels – **pragmatic, semantic, syntactic, and implementation**.

Second are the associations of elements that extend across the levels, which are termed as **Inter-Level CAD**. An example of such associations can be derived based on the relationship between previously mentioned organisational policy - ‘*recruiting of employees*’ and the operational business process that hires people. In the real-life process, there would be an action - ‘*recommends the selected candidate*’. The policy would detail, who, when, how and under what conditions this action can be performed. Similar to this **inter-level** association between **pragmatic and semantic levels**, there are two other associations between **semantic and syntactic levels**, and **syntactic and implementation levels**.

The main objective of understanding the **inter** and **intra-level** correlations is to capture them into analytical models that allow locating the propagating impact. A computer-based implementation of such a model would allow searching the model, to support the task of reflecting business process changes in already implemented web-based workflows.

In this view, this chapter covers the following;

- **Explores the nature of Intra-level CAD**, in the all four levels- pragmatic, semantic, syntactic and implementation
- **Creates models that can be implemented, to capture Intra-level CAD** in three levels- pragmatic, syntactic and implementation levels and
- Explores and create models to capture and implement Inter-level CAD between pragmatic and semantic, semantic and syntactic, syntactic and implementation

Creation of an implementable model for the semantic level CAD will not be covered in this chapter, due to its complex nature that needs further exploration.

This chapter contributes to two steps of constructive research; a) comprehension of existing theoretical body of knowledge and b) partial development of the solution. In order to achieve the above step a), exploration into both inter and intra-level CAD, is carried out in two forms. Firstly, certain scenarios drawn from the exploratory case study detailed in Chapter-4 are used. Secondly, from the research that has already studied certain elements of such inter and intra-level CAD.

There are mainly two sections in this chapter. The first section will discuss intra-level correlation of elements in the pragmatic, semantic, syntactic, and implementation levels. As mentioned before, these discussions at each level will extend to find a suitable data structures to capture intra-level CAD, except at the semantic level. The second section of this chapter will concentrate on finding inter-level CAD between pragmatic and semantic, semantic and syntactic and syntactic and implementation levels. Similar to the first section, suitable data schemata will be presented to capture the above-mentioned inter-level correlations.

5.2 Intra-Level CAD

The discussion of intra-level correlations will first investigate a specific example based on OCAS (Online Course Approval System) case study. Then these findings will be generalised to create a data schema to capture these correlations in any process.

5.2.1 CAD within Elements in the Pragmatic Level

The pragmatic level represents different contextual information artefacts that are both external and internal to the organisation (Acts, policies, organisational structures, goals and guidelines). These contextual information artefacts are by definition associated with each other.

These associations are sometimes clearly mentioned in the contextual artefact itself. For example, consider the following reference made in the clause 31 of
‘courses approval policy’ (UWS, 2003-2007b) in UWS to another policy. “(31) The College will convene the External Advisory Committee and present its report as part of the full proposal (See Course External Advisory Committee Policy)”(UWS, 2003-2007b, p6).

In certain other cases, the association among contextual information artefacts would not be that obvious. Consider the following part of clause 27 in the ‘courses approval policy’(UWS, 2003-2007b) in UWS. “(27) …… The Courses Officer informs the University, via the appropriate website, of the approval to proceed to a full course proposal.”(UWS, 2003-2007b, p5). In this case, the clause refers to a process participant named ‘courses officer’. The role ‘courses officer’ is identified in the management structure (UWS, 2004b) of UWS and role responsibilities and duties are defined in the general staff agreement (UWS, 2006-2008) of UWS.

Above examples, show a hierarchical association between the elements in the pragmatic level. The importance of understanding the policy hierarchies and models was identified by Wies (1995). In this work, he concentrates on a set of contextual information artefacts within the organisational boundaries. For this research, it requires a mechanism to extend the hierarchy beyond the organisational boundaries. Maullo (1993) is one of the researchers who has recognised relationships among contextual information that goes beyond organisational boundaries. Both Wies (1995) and Maullo (1993) fail to provide an unambiguous mechanism, to capture correlations among both internal and external contextual information artefacts, which defines a business process.

The following is the full list of contextual information artefacts (in the pragmatic level) related to the processes automated in OCAS:

i) Australian Federal Governments Higher Education Support Act 2003,
ii) Higher Education Provider Guidelines pursuant to the Act,
iii) UWS Strategic Plan and Vision,
iv) UWS Courses Approvals Policy,
v) UWS Governance Structure,
vi) UWS Management Structure,
vii) Courses and Unit Approval Timeline Document pursuant to the Policy,
viii) Course and Unit Templates and Guidelines pursuant to the Policy,
ix) UWS graduate attribute statements, 

x) Course External Advisory Committee Policy 

xi) Record Management Policy  

xii) Academic Staff Agreement  

xiii) General Staff Agreement

From the above list, contextual information sources i) and ii) are external to UWS, while others, iii) through to xiii), are created and managed within UWS. The courses approvals policy (UWS, 2003-2007b) is the core contextual information artefact that is directly related to processes automated in OCAS. The hierarchical relationships between above listed contextual artefacts are depicted in Figure 5.1 below.

The Courses approvals policy, similar to other policy documents is structured into sections and sub sections. Maullo et al. (1993) refer to sections and sub sections of a contextual information sources as fragments. These fragments generally have labels, which are numbered according systematic and agreed manner. For example, the clause (3).b in the courses approvals policy or statement 3 in the UWS vision statement.

The content text under a particular fragment carries information on the associations it has with other contextual artefacts. For example, the clause (3).b in
the courses approvals policy states, “The courses approvals process should ensure that courses support the achievement of UWS Graduate Attributes” (UWS, 2003-2007b, p3). This clause shows an association with the graduate attribute statement of UWS (UWS, 2003-2007c). This example of clause (3).b has only one relationship, which is directly mentioned in the descriptive clause. In addition, it shows the association with the whole statement of graduate attributes.

The association may be indirect as well as partial and multiple. There is an indirect association when a clause does not directly mention another contextual information artefact, but generally understood by the people who manage or use the policy. The partiality denotes, clause having a direct or indirect association with part of another contextual information artefact, but not to the whole document.

Multiplicity refers to one clause referring to more than one contextual information artefacts. Consider the clause (31) in courses approval policy, “The College will convene the External Advisory Committee and present its report as part of the full proposal” (UWS, 2003-2007b, p6). This particular clause associates two other contextual information artefacts as follows. First clause (31) has a indirect and partial reference to the Australian Federal Governments Higher Education Support Act’s Subdivision 19C (Australia, 2003), which describes the quality requirements of the organisation and the products (courses). Secondly, the procedures involved in convening the external advisory committee are described in the Course External Advisory Committee policy (UWS, 2003-2007a).

In summarising the above findings in relation to OCAS related contextual information artefacts, the types of associations that one contextual information element has with another can be classified into following:

- **Direct** – This is the case when fragments directly refer to another contextual artefact in the description.
- **Indirect** – This denotes the situation when the other contextual information artefacts are not directly mentioned, but generally understood by practitioners.
- **Partial** – This refers to one contextual artefact associated to a section of another contextual artefact, but not the whole document.
• **Full** – This refers to one contextual artefact associated with another contextual element in full.

• **Single** – This refers to one fragment making reference to just one other contextual artefact

• **Multiple** – This indicates one fragment referring to more than one contextual information artefact

Irrespective of the nature of the associations, it is imperative to record the correlations that exist among contextual artefacts, to approach the problem of evolution impact analysis holistically. Therefore, a data schema that supports the capture of the correlations among contextual elements at the *pragmatic level* is required to support the above-mentioned types of associations.

### 5.2.1.1 Schema to capture Intra-level CAD in the Pragmatic Level

Summarising the findings from the example from OCAS and the literature the following are listed to be the requirements that need to be fulfilled in a data schema that allows the capture of the CAD within the *pragmatic level*:

• Ability to capture direct, indirect, partial, full, single, and multiple associations made within contextual information fragments.

• Ability to show both hierarchical and non-hierarchical associations that may exist among contextual information artefacts (Maus, 2001; Wies, 1995).

• Ability to locate the contextual information artefact from a source document or a website where it is electronically published.

• Ability to refer to contextual artefacts in different formats.

In this research the requirement is that a mechanism to represent *intra-level* correlations at the *pragmatic level* and not to synthesise contextual information fragments to understand the semantics as proposed by Ramesh et al. (2005). Ramesh et al. (2005) present a knowledgebase system that recommends process changes after synthesising the contextual information changes. As they have correctly noted, these recommendations are required to be inspected by a human business analyst to assess the suitability. Thus, the use of descriptive language representation of policies so that
they can be synthesised, as done by Koch et al. (1996) and Kagal et al. (2003), is considered to be an excessive for the requirements of this research.

The contextual information referred to in this research, can come in a variety of formats, which cannot be easily synthesised. For instance, artefacts such as policies, laws and Acts, generally have a different format in comparison to organisational structures. This makes it complex to analyse these multi-format contextual information based on the content. Accordingly, the requirement here is to have a mechanism to refer to the fragments of the contextual information artefacts. In addition, some information that helps to locate the item physically, from websites or published documents, is also considered advantageous.

The schema in Figure 5.2 is proposed to capture the *intra-level* correlations at the *pragmatic level*. This schema collates information into five entities; i) `contxtInfoRegistry`, ii) `fragmentRegistry`, iii) `ContextAssociations`, iv) `fragmentAssociations`, and v) `fragmentContextAssociations`.

![Figure 5.2- Schema to Capture Contextual Information Associations](image)

- **Contextual Information Registry** (`contxtInfoRegistry`): This registry keeps a record of contextual information artefacts that are of particular importance to the business process. The attribute *context information ID* (`ContextInfoID`) is a unique number that identifies each context information artefact. The *context information type* (`typeOfContextInfo`) has enumerated data attributes, which

---

Chapter 5-Constraints, Associations, and Dependencies (CAD) of Process Elements
records type of a particular artefact such as a law, policy, Act, regulation and organisational chart. The *scope* attribute defines whether this artefact is applicable only within the organisation or external to the organisation. The *location* attribute allows capturing a URL or documenting a call number that helps to physically locate the contextual information artefact. *Effect from* and *expiry date*, are two date type attributes that capture the validity period of the contextual information artefact.

- **Fragments Registry** (*fragmentRegistry*): The fragments may refer to clauses, statements, paragraphs, or component parts of the contextual information. Every contextual information artefact may not be fragmented, for example organisational structures. However, in the contextual artefacts that can be fragmented, the following information is recorded: *fragment ID* that gives a unique number, *fragment label* captures some textual labelling such as clause 3.b or statement 1, *fragment name* records a descriptive name, and *contextual information ID* indicates which contextual information the fragment belongs to.

- **Contextual Information Association** (*ContextAssociations*): This entity allows the recording of full and multiple associations among contextual information artefacts. In this entity, the *CAID* attribute keeps a unique reference to the associations. The other two attributes allow a record of association between two full contextual artefacts.

- **Fragment Associations** (*fragmentAssociations*): Similar to previous contextual artefact associations above, this entity records the associations within fragments. This entity records the partial and multiple associations among fragments. Therefore, attributes record associated other fragments in foreign keys.

- **Fragment and Context Associations** (*fragmentContextAssociations*): This entity records the multiple associations between parts of one contextual artefact to fragments of another. Similar to previous entity foreign keys are used to record the associations.

Table 5.1 analyses the above schema’s (Figure 5.2) ability to achieve the present intended outcomes.
Table 5.1- The ability of the Schema in Figure 5.2 to Capture Correlations among Contextual Information Elements

<table>
<thead>
<tr>
<th>REQUIREMENT</th>
<th>METHOD TO FACILITATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supporting Direct and indirect associations</td>
<td>It requires a human business analyst to identify both direct and indirect associations, as the content within the contextual information is not synthesised in this model. Therefore, at the inception when this schema is populated, it is manually identified and entered.</td>
</tr>
<tr>
<td>Partial and full associations</td>
<td>breaking the contextual information artefact into fragments and maintaining the associations in different associative entity</td>
</tr>
<tr>
<td>Single and multiple associations</td>
<td>- maintaining the associations in different associative entity</td>
</tr>
<tr>
<td>Hierarchical and non-hierarchical</td>
<td>The hierarchical association is shown using the self reference in the context registry Non-hierarchical association is in the associative entity</td>
</tr>
<tr>
<td>locate the contextual information</td>
<td>The attribute location facilitate this</td>
</tr>
<tr>
<td>support contextual artefacts in different formats</td>
<td>- Facilitated by only keeping references to important component parts, rather than keeping the content</td>
</tr>
</tbody>
</table>

The schema given in Figure 5.2 captures the correlations among elements at the pragmatic level is one of the components (among few others) which later contribute to the Process Evolution and Change Impact Analysis (PECIA) Model, in Chapter-8.

### 5.2.2 CAD within elements in the Semantic Level

Semantic level represents the business processes in practise in organisations.

A process has three dimensions (elements) - actions, participants, and object. In this section, initially associations and dependencies in relation to process elements are explored. Next, constraints that affect the above identified associations and dependencies are examined. This section expands under following sub headings:

- Associations and Dependencies within Process Actions
- Associations and Dependencies between Process Actions and Participants
- Associations and Dependencies between Process Actions and Business Object
- Associations and Dependencies within Business Object
- Constraints on the associations within Process Actions
- Constraints on the associations between Process Actions and Participants
- Constraints on the associations between process Actions and Business Object
- Constraints on the associations within Business Object Elements
By combining the findings under above headings, finally in this section, the **CAD model of process elements** is developed.

### 5.2.2.1 Associations and Dependencies within Process Actions

One of the core aspects of processes is its association within actions or work tasks (van der Aalst et al., 2003b). The literature studies the types of associations that actions can have among them, in detail.

<table>
<thead>
<tr>
<th>Relation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>x before y</td>
<td><img src="image" alt="Sequence" /></td>
</tr>
<tr>
<td>x meets y</td>
<td><img src="image" alt="Parallel Split" /></td>
</tr>
<tr>
<td>x overlaps y</td>
<td><img src="image" alt="Synchronisation" /></td>
</tr>
<tr>
<td>x starts y</td>
<td><img src="image" alt="Exclusive Choice" /></td>
</tr>
<tr>
<td>x during y</td>
<td><img src="image" alt="Simple Merge" /></td>
</tr>
<tr>
<td>x finishes y</td>
<td><img src="image" alt="Discriminator" /></td>
</tr>
<tr>
<td>x equals y</td>
<td><img src="image" alt="Merge" /></td>
</tr>
</tbody>
</table>

*Figure 5.3- Temporal Intervals between two Elements – adopted from Allen (1983)*

Allen’s (1983) work (Figure 5.3) in interval-based temporal logic is one of the early studies that gives a notion of a pattern between two items (in this case two actions) in the space of time. He identifies seven patterns to represent the relative positions between pair of items (say x and y).

While Allen’s (1983) temporal patterns can be used to represent the relationship between two actions, van der Aalst el al’s (2003b) twenty (20) workflow patterns gives a much more refined and applicable framework to map other factors beyond two actions. In this work, they identify six categories to define the associations among actions:

- Five (5) basic process flow patterns- *sequence, parallel split, synchronisation, exclusive choice* and *simple merge*,
- four (4) advanced branching and synchronisation patterns - *multi-choice, synchronising merge, multi-merge* and *discriminator*,
- two (2) structural patterns - *arbitrary cycles* and *implicit termination*,
- four (4) patterns involving multiple instances - *multiple instances without synchronisation, multiple instances with a priori design time knowledge, multiple
instances with a priori runtime knowledge and multiple instances without a priori runtime knowledge,

- three (3) state-based patterns - deferred choice, interleaved parallel routing and milestone, and
- two (2) cancellation patterns - cancel activity and cancel case.

The work of van der Aalst (2003b) was later extended by Russell et al. (2006a) into 43 patterns. The new 23 patterns are introduced to clarify certain ambiguities and issues that existed in the previous 20 patterns. However, in this extension the basic flow patterns have remained to be the same.

In addition to theoretical studies (Russell et al., 2006a; van der Aalst et al., 2003b; Allen, 1983) practical usage of associations among actions can be extracted from process modelling tools. Business Process Execution Language (BPEL) also known as BPEL4WS (BPEL for Web Services) is such process definition language (Andrews et al., 2003). In BPEL specifications, there are a number of flow patterns introduced. The main flow patterns include the following: sequence (<sequence>), parallel split (<flow>), conditional choice (<switch>), looping (<while>) and skipping of an action (<skip>). The element shown in the parenthesis shows the BPEL construct to represent the control flow patterns.
Figure 5.4- Flow Patterns Identified in the New Course Approval Process in OCAS
A practical use of control flow patterns can be further exemplified using the processes associated with the OCAS case study. In OCAS’s, New Course Approval Process supports sequence, parallel split, exclusive choice, simple merge, and multi-choice flow patterns. In addition, iteration of actions is also present in the OCAS related processes. The process diagram, of the New Course Approval Process (Figure 5.4) highlights the occurrences of flow patterns in OCAS processes.

Based on the numerous flow patterns identified in the literature and used in OCAS processes, the sequence, parallel split, conditional split, synchronisation, simple merge, iteration and skip are the most common flow patterns that exist among process actions. This is also verified based on the survey findings of seven workflow products carried out by van der Aalst et al. (2003b) and Russell et al. (2006a). According to their survey results, sequence, parallel split, synchronisation, exclusive choice, simple merge, multi-choice are the only ones that are directly supported by all surveyed commercial workflow products. In addition, when comparing with Allen’s (1983) patterns, it can be observed that the sequence, split (parallel or conditional) and merge (simple or multiple) appears to be the core types of patterns.

Figure 5.5 concisely summarise the above discussion in presenting the common types of associations that can exist among process actions. These associations among process actions are symbolised using the notation f(A), denoting a function of actions.

Figure 5.5 contributes to creating the CAD model of process elements presented at the end of this section.
5.2.2.2 Associations and Dependencies between Actions and Participants

In this thesis, the term ‘participant’ refers to human participants who are identified based on their organisational roles. These participants identified in organisational roles are associated with process actions. There are a number of research studies that examine these associations between process participants and actions.

Kagal et al. (2003) identify four types of association among users and systems as - rights, obligations, prohibitions, and dispensations. Rights refer to permissions certain participants have to perform particular actions. Obligation is the mandatory requirement of one action to be performed by selected participant/s. As the name suggests the prohibition refers to certain process actions not being allowed to be performed by participant/s. Dispensing refers to a deferred choice in associating a participant with an action.

Russell et al.’s (2005b) work on resource patterns is one of the other comprehensive studies that investigates the associations between process participants and actions. In this study, the authors classify 43 patterns under seven categories. This study provides for a framework to understand patterns of associations between process participants and actions. However, in this work the focus is more on constraints on the associations such as capacity based allocation, organisational allocation and allocation based on history of performing action, rather than the types of associations. Russell et al.’s (2005b) study into the constraints, is discussed later in this section.

In OACS related processes, there are over thirty (30) roles identified to perform over 40 actions. When analysing the associations between the participants and actions in OCAS related processes, it is possible to locate both permissions and obligations occurring frequently. These obligations and permissions in OCAS are explained in the following examples.

Consider the two actions; Submit Course Notification and Recommend Course Notification, presented in the high-level process diagram format in Figure in 5.6 and state table representation for those actions in Figure 5.7. Chapter-4 presented bubble charts (Figure 5.6) and state tables (Figure 5.7) as the high-level and detailed process modelling tools used in OCAS project.
Figure 5.6- High-level Process Diagram for Two Actions

Figure 5.7- State Table Representation for the Two Actions in Figure 5.7

<table>
<thead>
<tr>
<th>current state</th>
<th>entry condition</th>
<th>automated actions on entry</th>
<th>actor</th>
<th>participant performed actions</th>
<th>form events</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-notification_being_created</td>
<td>null</td>
<td>null</td>
<td>project_manager</td>
<td>submit_course_notification</td>
<td>save</td>
</tr>
<tr>
<td></td>
<td>null</td>
<td>null</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3-notification_waiting_hos_decision</td>
<td>null</td>
<td>{name=&gt;&quot;send_email&quot;,arg=&gt;{&quot;to&quot;:&quot;course_officer&quot;,&quot;template&quot;:&quot;new_course&quot;}}</td>
<td>head_of_school</td>
<td>recommend_course_notification</td>
<td>approve</td>
</tr>
<tr>
<td></td>
<td>null</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>null</td>
<td></td>
<td></td>
<td></td>
<td>reject</td>
</tr>
<tr>
<td></td>
<td>null</td>
<td></td>
<td></td>
<td></td>
<td>re_submit</td>
</tr>
<tr>
<td>onshore=1</td>
<td>null</td>
<td></td>
<td>dean</td>
<td>deans_executive_veto_for_course_notification</td>
<td>reject</td>
</tr>
<tr>
<td></td>
<td>null</td>
<td></td>
<td>academics</td>
<td>view_course_notification</td>
<td>null</td>
</tr>
</tbody>
</table>

Figure 5.6 shows the direct association of process participants, in the roles of proposer and Head of School, with the two actions Submit Course Notification and Recommend Course Notification respectively. These participants have certain obligations to perform these actions, as the process is unable to move forward without these participants performing these actions.

In the state table representation in Figure 5.7, there are additional actions other than Submit Course Notification and Recommend Course Notification in the related states. These additional actions are Dean’s executive veto for course
notification and view course notification. The participants who are associated with those actions are Dean and Academics respectively. However, in relation to these additional actions, the participant is permitted to perform the actions and their inaction does not create issues with the process flow.

OCAS does not present any prohibitions explicitly in process models. This is due to the reason that the modelling approach used is similar to the ‘default deny’ concept of setting authentication rules in firewalls (Zwicky, Chapman, & Cooper, 2000). This means that everyone’s access by default denied unless specified otherwise. However, in cases in which it requires the ‘default permit’ approach to be used, it would be beneficial for forbiddance associations to be present to prohibit access to specific personnel.

Figure 5.8 succinctly summarises the above-discussed types of associations - obligations, permissions and prohibitions (or forbiddances) that could exist between process participants and actions.

![Figure 5.8- The Types of Associations between Process Participants and Actions](image)

Here the notation f(A,P) symbolises a function of actions and participants that would be required to be presented in a formalism. Figure 5.8 contributes to the CAD model of process elements developed at the end of this section.

### 5.2.2.3 Associations and Dependencies between Actions and Business Object

Object or data is an important dimension in today’s business processes. The early research in relation to workflow modelling did not pay much attention to
capturing the data aspect. Identifying this gap, there are several proposals (Lehmann, 2006; Sadiq et al., 1999b) for capturing data aspect of workflows.

In web-based workflows, actions are closely associated with *informational business objects*. Generally, participants perform actions using system user interfaces (UIs). In web-workflows, these UIs tend to be either in the form of documents (HTML, PDF, Word), interactive web forms, or a combination of both. The data that populate these UIs, are both internal and external to the workflow system and comes from sources such as relational or object oriented databases, files in variety of formats (PDF, XML, Excel, Word) and/or XML databases (Lehmann, 2006).

In 2005 (Russell et al.) a comprehensive study was carried out to explore the data patterns in workflows. In this work, Russell et al. (2005a) studied workflow data patterns in four categories as follows;

- **Data visibility**: defines the patterns in which data can be visible to an action or block of actions or multiples instances.
- **Data interaction**: defines the ways in which data elements are transferred between actions, blocks of actions and instances.
- **Data transfer mechanisms**: this refers to the implementation mechanisms, such as transferring data by value or by reference, to support the above-discussed data interactions. The data transfer mechanisms are not directly associated to the topic under discussion; therefore, this factor is not further discussed.
- **Data Based Routing**: this refers to ways in which data is used for process routing purposes. This important association exists between process actions and information object of the process.

The main process object in the OCAS related *New Course Approval Process* is a course. In this information object, there are over 200 attributes or data elements. This data is maintained in an object-relational database supported by Liang and Ginige's (2006) Smart Business Object concept.
Consider the example UI from OCAS in Figure 5.9, which allows participants to perform a particular process action. According to the labelling in the figure, (i) shows a link to where the participant is allowed to access a portion of the business object data in the report format. The labels (ii) and (iii) show an attribute that the participant is allowed to edit. As previously mentioned, a course business object has over 200 attributes. However, from those attributes, the UI given in Figure 5.9 will allow the user to edit only two elements. This shows the visibility trait identified in Russell et al.’s (2005a) work.

The visibility of attributes in action UIs in OCAS defined in following based on the parameters given in Table 5.2.
Table 5.2- Visibility Aspect of Business Object Attributes in an Action UI

<table>
<thead>
<tr>
<th>VISIBILITY TRAIT</th>
<th>ADDITIONAL OPTIONS</th>
<th>EXAMPLE / DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viewable</td>
<td>Report</td>
<td>Here the attribute is only allowed to be viewed, but can be viewed in a report or on the web-form itself. It also needs to define: - location of display (page number, screen number) and - format of display (mm/yy, Mon/Year)</td>
</tr>
<tr>
<td></td>
<td>On Screen</td>
<td></td>
</tr>
<tr>
<td>Editable</td>
<td>Form control type</td>
<td>Editable data is presented in an appropriate form controls are required. These include: - text box - selection box - radio buttons - check boxes</td>
</tr>
<tr>
<td></td>
<td>Validation details</td>
<td>Validations on the data collected these would specify: - Mandatory or Non-mandatory nature - Format - Length - Allowed and denied characters</td>
</tr>
<tr>
<td>Hidden</td>
<td></td>
<td>Indicates when data is hidden from an action UI</td>
</tr>
</tbody>
</table>

Data collected through the UI similar to the one in Figure 5.9 is also used for making routing decisions. Consider the example given in Figure 5.10. This shows the final approval of business case (BC) of the New Course Approval Process. Here the action is assigned to different participants based on some values in the three attributes exception, offshore and onshore of the course object. These attributes are populated in another action UI. This example shows the concepts of data interactions and data-based routing by Russell et al. (2005a).

Figure 5.10- Data-Based Routing in OCAS

Figure 5.11 summarises the types of associations that exists between action UIs and business object data.
In Figure 5.11 the $f(A,D)$ symbolises the function of actions and data elements, that is presented in a formalism later in Chapter-6. This Figure also contributes to the CAD model developed at the end of this section.

### 5.2.2.4 Associations and Dependencies within Business Objects

Business objects associated with business processes could comprise either; physical objects (personal computer in a computer assembling process) or informational presentation of a physical object (loan in a loan approval process) or a combination of both physical object and informational objects (Malhota, 1998). In this thesis, the focus is on information objects related processes. There are several methods to represent process related information objects. For example, it could be in the form of a document (for example PDF or Word) or much more structured XML file or in a relational or object oriented database (Lehmann, 2006).

The data elements or attributes in an informational business object are related to each other in two ways. These are integrity and computational relationships:

- **Integrity Relationships** – refers to the situation when one data entity in a database or a document is linked with another, to create a composite informational object (Whitten et al., 2007a; Whitten et al., 2007b; Banerjee et al., 1987). Generally, in a DBMS, the integrity relations are maintained via primary key and foreign key relationships. In documents, these kinds of associations are controlled via document links or hyperlinks.

As mentioned previously the business object of OCAS processes have over 200 attributes. For easy management, these elements are separated into tables.
and documents, and then appropriately linked to maintain integrity relationships.

- **Computational Relationships** – In an information object there are certain composite process attributes such as summations, averages, percentages and any other value calculated using mathematical or non-mathematical function. These attributes are generally referred to as derived attributes (Whitten et al., 2007b). These derived or computed attributes based on values of other attributes is the second type of relationship that exist within business object. In OCAS the data attributes such as total estimated cost or expected revenue are calculated based on variety of other data elements, such as resource costs, student fees, student numbers and taxes.

![Figure 5.12- Associations among Data attributes of a Business Object](image)

The above-identified associations among data elements are presented in Figure 5.12. In this figure, f(D) refers to the formalism that defines the integrity and computational dependencies, which will be further explored in Chapter-6. Further, similar to previous models; Figure 5.12 also contribute to the CAD model of process elements presented later in this section.

### 5.2.2.5 Constraints on the Associations within Process Actions

In processes, actions are ordered systematically to achieve the business goals. However, there are always exceptions that take place that prohibits actions taking place as planned. Therefore, the associations among process elements are constrained on the occurrence or non-occurrence of these exceptions.

In the series of studies that examined workflow related patterns⁹, these exceptions are studied by Russell et al. (2006b). In this work, they identify five types

---

⁹ This work is carried out in the name of Workflow Pattern Initiative. The related work is published at [http://www.workflowpatterns.com/](http://www.workflowpatterns.com/)
of exception patterns namely: work item failures, deadline expiry, resource/data unavailability, external trigger, and constraints triggers.

Although, Russell et al. (2006b) have identified the deadline expiry as one of the exceptions, this aspect was initially studied under the term time or temporal constraints by Marjanovic (2000). In this work, she presents the concept of absolute and relative deadline constraints in production workflows. In addition, researchers such as Sadiq et al. (2000a) and Lu et al. (2006c) show the effect of time in creating, allocating and completing process action.

![Figure 5.13- External Triggers (Rejection of BC) affecting other Processes](image)

From the above exception patterns, the external trigger is the only exception pattern explicitly supported in OCAS. In OCAS processes, rejection of a business case (BC) creates an external trigger to stop all the sub processes associated with course approval. This scenario is extracted and presented in Figure 5.13. In this case, the rejection of the BC by the Pro -Vice Chancellor Academic, forces all the other sub-processes to move into their own rejection state.

By summarising the above discussion, the work item failures, time constraints, resource/data unavailability, external trigger, and constraints triggers are considered as the set of constraints that affect the association among process actions (Figure 5.14).
In Figure 5.14 the notation $C[f(A)]$ symbolises all constraints that affect the association among process actions - $f(A)$. This is the first type of constraint that contributes to the CAD model of process elements.

### 5.2.2.6 Constraints on the Associations between Actions and Participants

Association of actions with participants can be constrained by a number of factors. Russell et al. (2005b) identify most of these factors in their work on resource patterns. In addition, researchers such as Marjanovic (2000), Sadiq et al. (2000a) and Lu et al. (2006c) also identify certain factors that affect the allocation of process tasks to participants.

These factors that constrain action association to participant can be broadly categorised into three; i) certain personal characteristic(s) possessed by an individual participant, ii) arise from a relationship between two participants and iii) external factors. The following lists are extracted from the studies by Marjanovic (2000), Sadiq et al. (2000a), Ferraiolo et al. (2001), Russell et al. (2005b) and Lu et al. (2006c).

i) Personal characteristics

- **Location**- refers to allocation of a task based on the location of the participant (Russell et al., 2005a).
- **Experience**- denotes the allocation of a task being constrained by experience or inexperience of a participant (Russell et al., 2005a).
- **Skills or capabilities**- refers to situations when association of actions with participants being constrained by the skills or the capabilities of the participant (Russell et al., 2005a).
• **Availability**- On some occasions the association of a participant is constrained by the availability of the participant, during a specific time.

• **Workload**- In certain processes, the workload of the participant is taken into consideration, when allocating tasks (Russell et al., 2005a).

• **History**- On certain occasions, history in performing past process actions would be required to be considered before allocating new tasks to a participant (Russell et al., 2005a).

• **Preference**- Some actions could be allocated to a participant depending on the preference or the interests of the participants (Russell et al., 2005a).

• **Belonging to a sub-structure**- Usually, participants are identified in organisational roles. These roles are grouped into sub structures, generally referred to as organisational units or groups. When allocating tasks to a participant, this belonging to a sub structure of an organisation could be considered (Russell et al., 2005a; Ferraiolo et al., 2001).

• **Level in the organisational role**- In addition to the role held by the participant, participants organisational level can constraints the association with actions. This is particularly applicable in the case of where multiple roles exist in different levels, such as Academics in Higher Education Workplace (HEW) level A to E (Russell et al., 2005a; Ferraiolo et al., 2001).

ii) Relationship between two participants

• **Belonging to same and different sub-structure**- One task can be allocated to a participant based not only on his/her belonging to an organisational unit, but also in comparison with another participant’s belonging to an organisational unit (Russell et al., 2005a). For example, *the final payment of the claim to an employee needs to be approved by a manager who belongs to the same department as the employee*.

• **Reporting and Delegation Authorities**- The association of an action to a participant can be constrained based on the participant’s reporting and delegation permission. For example, the ‘*manager of the employee needs to approve the leave form’*(Russell et al., 2005a).

• **Inter personal relationships**- In some processes the informal interpersonal relationships among participants need to be taken into consideration when
allocating process tasks. For example, the human resources manager is not allowed to interview a close family member for a job position.

iii) External Factors

- **Time constraints**- One of the common external factors that affect the association of process participant with actions are the time constraints. For example, a deadline expiry might suggest escalating a task to a different participant. Also in some processes, absolute time would be taken into consideration (Russell et al., 2005a; Sadiq et al., 2005; Marjanovic, 2000). For example, after 8.00 am, cleaners are not allowed to perform cleaning duties of office rooms.

- **Method of allocation**- Sometimes, the technique used in the business process to allocate actions to participants also could constrain associating actions with participants. These techniques or methods such as round robin, random, schedule based or piled allocation are the common methods of allocations (Russell et al., 2005a). For example, front office duties are allocated to each employee based on a roster (scheduled allocations).

In OCAS related processes, only a few of the above constraints are used in allocating process actions to participants. The most commonly used constraints are the reporting and delegation authorities, belonging to same or different organisational sub-structures, level of the organisational role, workload and experience. For example in the New Course Approval Process, when both Head of School and Dean are required to perform approvals, generally the Head of School approvals are prior to the Dean’s approval. This is for the simple reason that heads of schools report to Deans. Another example is, when an Academic proposes a course in OCAS, the Head of School and Dean are selected for subsequent approvals. This selection is based on the participants belonging to the same school and college as the Academic.
In summary, the constraints that could affect the association of actions with participants are denoted in Figure 5.15. In this Figure the notation \( C \{f(A,P)\} \) symbolises these constraints listed.

### 5.2.2.7 Constraints on the Associations between Actions and Business Object

The association between the actions and business object data was previously identified in three forms as data visibility, interaction, and data-based routing. The constraints described below are mostly, on the visibility trait. The visibility of certain business object elements could be constrained by factors such as *ownership of data* and *specialisation of data*.

- **Ownership of data** – Information that is stored in business objects can belong to a particular participant, group of participants, to an organisational unit or to the organisation. Depending on this ownership, the visibility of the data in actions can be constrained. For example, leave application of an employee could be only visible to that applicant him/herself, the manager of the applicant and the leave clerk.

- **Specialisation of data**- An example of specialisations of business object is *post-graduate doctoral research course* as oppose to any other courses in the university. This kind of specialisation would constrain the visibility of certain data elements in action interfaces.

In OCAS related processes, the ownership of a course by a particular school or college decides the visibility of that data to certain participants. For example, *Dean* is only allowed to approve the courses that belong to his/her *college*. 
Figure 5.16- Constraints on the Association among Process Actions and Business Object

Figure 5.16 depicts the constraints that affect the association of data with action interfaces as $C\{f(A,D)\}$. These constraints are specifically on the visibility trait of the association.

5.2.2.8 Constraints on the Associations within Business Object Elements

The types of associations among business object elements were previously identified as integrity and computational relationships. Specialisations are considered the only type of relationship that constraints these associations. Similar to the above, a specialisation of a business object can be exemplified by home or personal loans in loan approval systems. Based on the specialisations computational relationships can change.

Figure 5.17- Constraints on the Association among Business Object Elements

Figure 5.17 depicts the constraints on the associations among object element associations, as $C\{f(D)\}$. 
5.2.2.9 Model of CAD among Process Elements

In sections prior to this, four types of associations and dependencies were identified as follows:
- Associations and Dependencies within Process Actions – \( f(A) \)
- Associations and Dependencies between Process Actions and Participants – \( f(P,A) \)
- Associations and Dependencies between Process Actions and Business Object – \( f(A,D) \)
- Associations and Dependencies within Business Object – \( f(D) \)

Followed by the identification of the above associations and dependencies a set of constraints were discussed as follows:
- Constraints on the associations among process actions – \( C\{f(A)\} \)
- Constraints on the associations between process actions and participants – \( C\{f(A,P)\} \)
- Constraints on the associations between process actions and business object elements - \( C\{f(A,D)\} \)
- Constraints on the associations among business object data elements - \( C\{f(D)\} \)

The above-discussed, associations, dependencies and constraints are brought together in the model shown in Figure 5.18.
Although it is not specifically mentioned, the above CAD model of process elements is similar to a meta-model that attempts to conceptualise the types of associations among process elements. Previously in Chapter-3 a number of meta-models were discussed, including; Workflow activity model –WAMO (Eder et al., 1995), Dynamic Workflow Management – DWM meta-model (Kwan et al., 1997), ADOME-WFMS meta-model (Chiu et al., 1999), WfMC’s meta-model (WfMC, 1999) and Event-driven Process Chain (EPC) Meta-Model (Stefanov et al., 2005).

The CAD model of process elements (Figure 5.18) conceptually captures the same information presented in the above-mentioned meta-models. However, the major difference of this model is its distinction of associations and dependencies into one set of correlations and constraints into another. This separation allows the constraints to be modelled using an appropriate constraint modelling language.
The CAD model of process elements (Figure 5.18) presented here is only a diagrammatic representation. As mentioned in the chapter overviews, this section does not concentrate on developing a schema appropriate for capturing CAD among process elements. This is due to the inherent complexity in correlations between process elements. In the author’s view, simple entity-relationship data structures are not appropriate to conceptualise such complex structures that would facilitate deeper analysis required by this research. Therefore, the recommendation is to use a formalism similar to Process Algebra (Fokkink & Zantema, 1994). The decision on a suitable formalism and its application to capture CAD among process elements, leads to an extensive discussion. Therefore, exploration for a suitable formalism is detailed separately in Chapter-6.

### 5.2.3 CAD within elements in the Syntactic Level

The syntactic level of the PoPA framework represents the models created in system implementation. In general, there are three types of models created in system development. These are process model(s), business object or the data model(s), and interface definition(s) (Whitten et al., 2007a).

There can be variety of tools used to conceptualise the three main aspects of an application: processes, data, and interfaces. For example, the visual process models can be created using high-level process modelling tools such as UML activity diagrams (OMG, August 2005), Petri-Net diagrams (Working-Group, 1997) or Event-driven Process Chains –EPC (Mendling et al., 2005a). Then, these high-level process diagrams will be converted into more detailed machine-readable formats such as XML Process Definition Language – XPDL (XPDL, 2005) or Business Process Execution Language-BPEL (Andrews et al., 2003). The data models will be usually created either as entity relationship diagrams or class diagrams (Whitten et al., 2007a). Interface definition models could be developed using less formal methods such as storyboarding or rapid prototyping (Whitten et al., 2007b). The selection of an appropriate set of tools depends on a number of factors such as, development approach, familiarity with tools by development team and the ability of end-users and stakeholders to comprehend the models created using the selected tools.
The models created in process automation do not exist in isolation and are interlinked to produce the desired system outcome. For example, data captured in an object model would be used to populate the interface definitions. Hence, a change in an underlying data structure creates issues with the interfaces, unless these are appropriately handled. Therefore, it is imperative to understand the associations among process models in order to facilitate the evolution of process models at the implementation level of the PoPA framework.

Computer aided software design tools such as Rational Rose\textsuperscript{10} and Eclipse\textsuperscript{11} provide extensive frameworks to create and manage system related models. Use of such tools facilitates managing associations only among some system models. For example, these tools does not support creating interface definition models or managing its associations with other models.

The requirement of this research is to record and manage simple associative relationships that exist among models created in process automation. In order to comprehend such associations, first some of the models created in the OCAS project are analysed. The following models were created to capture the process elements and their associations in OCAS implementation.

- **Entity Relationship (ER) models**: There are two ER models used in OCAS
  - **Business Object Models**: The two business objects used in OCAS related processes are the *course* and *unit*. Two ER models were created to capture these two business objects with over 200 attributes in each of them.
  - **Application specific data**: OCAS uses a workflow management system (WFMS) to enact the process and glue front end application to actions interfaces (van der Aalst et al., 2002a). However, OCAS being a full process-oriented information system accommodates other peripheral information such as home page information, links to external policy documents, help information, frequently asked questions (FAQs) and any other control data required by the application. To manage this application data, OCAS uses an application specific database.

\textsuperscript{10} Rational Rose Software Design Tools \url{http://www-306.ibm.com/software/awdtools/developer/rose/} last accessed 17/01/07
\textsuperscript{11} Eclipse software modelling tools \url{http://www.eclipse.org/} last accessed 17/01/07
• **Process models**: There are two process models used in OCAS.
  o High-level process model in **bubble charts** - These are visual models conceptualise processes at a high-level.
  o Detail process model in **state tables** - These are detailed process models that capture information on alternate routes, non-compulsory actions, web-form action controls and automated actions such as sending e-mails, which are not present in high-level bubble charts. Details on bubble charts and state tables were previously discussed in Chapter-4.

• **User interface (UI) definition models**: These define UI for each action associated with processes automated in OCAS. In particular, the actions that are presented to humans as web-forms, need to clarify the visibility (view, edit and hidden) of data elements. Details on action UIs models were discussed in Chapter-4.

• **Participant model**: In OCAS, roles and organisational units are mapped into a model, to comprehend the control scope of a particular role. For example, the role *course officer* and *timetabling officer* is applicable at the *college level*. Therefore, there are three *course officers* and three *timetabling officers* for each college. However, the role *courses approval committee officer* is applicable at the *university level*. Therefore, there is only one *courses approval committee officer*. Similar to the above it requires specifying the scope of each process participant in relation to the organisational units of UWS. A model created to capture the above information is used and referred to as a participant model.

• **Storyboard for the web application**: OCAS is a process-oriented web application that also contains some non-process related web pages. These could range from static pages to dynamic pages generated using application specific data. Thus, it requires showing the ordering of these types of pages along with the UI dedicated for performing process actions. Therefore, a storyboarding model is used for this purpose.

  The correlation between these models created in OCAS project is depicted in Figure 5.19.
Initially the high-level process diagrams (bubble charts) and participant models were created together. Once high-level process in bubble chart is complete and agreed on, the detailed state tables are developed. Simultaneous to the process models, business object models (data models) are produced. UI models are designed based on the process models and data models, by matching actions to data objects. Finally, the application storyboard is created using action interface definitions and application specific information.

The above-mentioned models and their correlations are specific to the OCAS implementation. Another process automation effort may have different models and associations among models.

When analysing the correlations among the models two types of associations are evident.

- Inheriting associations – Inheriting associations refer to cases where model(s) required to be fully completed before creating another. This is because the second model fully inherits elements from the first one. A solid line in Figure 5.19 denotes this. An example would be Object models and State tables having to be completed before the creation of UI definitions of process actions.
• Referencing associations – This denotes the case, where the two models refer to each other for certain elements and definitions. Dotted lines shown in Figure 5.19 indicate these referential associations.

In this research, the requirement is to reference the appropriate models and record the relationships among them. In other words, it does not attempt to fragment model elements and record the correlations at element level. A schema that supports such information to be recorded should have the following capabilities within it:

• Ability to reference a model in any format (digital or paper).

• Ability to locate the model physically.

• Ability to show both inheriting and referencing correlations among models in a manner that it can identify the impact of change in one model to another.

Models created in automation exist in different forms; therefore, it is adequate to use a schema that allows keeping references with models, similar to the pragmatic level. The data structure given in Figure 5.20 allows referring to all types of models created that are related to a particular process automation exercise. In this referencing, it also records the many-to-many relationships that could exit between models, similar to with the OCAS implementation (Figure 5.19).

![Figure 5.20- Data Structure to Capture References to Models and their Associations](image)

The attributes `model1` and `model2` in `modelAssociation` entity are foreign keys of the models listed in the `modelRegistry`. The `typeofAssociations` defines the way in which two models can be related – inheritances and references.
This schema meets the needs previously set. First, it is able to keep references with all formats of models. It is also possible to locate the models, using the location attribute. In addition, it is able to distinguish between inheriting and referencing associations using the typeofAssociations attribute.

The schema presented in Figure 5.20 contributes to the PECIA Model presented in Chapter-8.

5.2.4 CAD within Elements in the Implementation Level

The Implementation level, in PoPA framework, signifies the implemented web-based workflow. A web-based workflow consists of a number of artefacts including:

- **Data stores:** Ranging from documents, relational or object oriented databases or XML databases (Lehmann, 2006).
- **Function logic:** In code files related to a web-based application in CGI scripts such as Perl or Java and embedded scripts such as java scripts, PHP and ASP.
- **Presentation logic:** In style files such as CSS and XSLT.

Depending on the chosen development approach, the above artefacts could be interlinked in a number of ways. A detailed study into such correlations is outside the scope of the main objectives of this research. Therefore, similar to pragmatic and syntactic levels, the focus is to maintain references to the implementation artefacts and record the possible correlations among them.

A schema that captures the references to web-artefacts and their associations would have the same requirements as for the schemata that are presented in the syntactic level. The schema given in Figure 5.21 is proposed for recording artefacts references and their associations at the implementation level.

![Figure 5.21- Schema to Capture Implementation Level Artefact Associations](image.png)
Similar to previous schemata here the artefact registry keeps a record of all elements that are associated with a web-based workflow. The model presented in Figure 5.21 also later contributes to the **PECIA Model** presented in Chapter-8.

This section (5.2) investigates different types of associations that may exist within the four levels of information representations – *pragmatic, semantic, syntactic* and *implementation*, in the **PoPA framework**. In this discussion, first elements of each level are identified and then their correlations are studied. This study is backed up by a literature review and OCAS specific examples.

The outcome of this study is the development of a set of schemata that allow capturing references to elements at each level and correlations among them. Schemata were developed for *pragmatic, syntactic, and semantic levels*. The next section of this chapter concentrates on studying **inter-level CAD** in relations to the four levels in the **PoPA framework**.

### 5.3 Inter-Level CAD

Inter–level CAD refers to correlations that elements in one level have with elements at another. Therefore, the *inter-level CAD* is studied under three sub-headings as follows:

- CAD between elements in the **pragmatic and semantic levels**
- CAD between elements in the **semantic and syntactic levels**
- CAD between elements in the **syntactic and implementation levels**

Similar to the previous study, in *intra-level element CAD*, this discussion also develops suitable data structures to capture CAD between elements at different levels.

#### 5.3.1 CAD between Elements in the Pragmatic and Semantic Levels

This section develops a data schema to facilitate the capturing of the associations between elements in the **pragmatic level** (contextual artefacts) and **semantic level** (process elements). Such a schema captures these associations cohesively, in order to facilitate propagating impact analysis, which flows from the **pragmatic level** down to **semantic level**. Prior to the introduction of the schema, first
the nature of associations between contextual artefacts and process elements are discussed.

Process elements (actions, participants, and business object elements) and the associations among them (previously discussed under section 5.2.2) are defined by business rules present in contextual artefact at the pragmatic level. Business rules define and control the lifecycle of products and services and the supporting infrastructure (Ali, Torabi, & Soh, 2007; Gottesdiener, 1997; Hay et al., 1997). In an organisation, there can be many thousands, if not hundreds of thousands of business rules. A sub-set of these rules helps to define business processes of that organisation (Ali et al., 2005a; SEEC, 2003). These applicable business rules are derived from the contextual artefacts such as policies (Hay et al., 1997) that are represented in the pragmatic level of the PoPA framework.

In the GUIDE business rules project, Hay and Healy (1997) identify four types of business rules as:

- **Definitions of business terms**: A term definition is a word, phrase, or sentence(s) which has a specific meaning for the business, which may be written as ‘Head of School is defined as ....’.

- **Facts relating terms to each other**: Asserting an association between two or more terms is a ‘fact relating term’. For instance, ‘each course should have a course coordinator’ is an example of relating facts.

- **Action assertions or constraints**: Constraints are circumstances under which a transaction, decision, action, or business occurrence should be aborted, incomplete, rejected, or undesirable. For example, ‘course cannot be approved if all resource divisions have not signed’ is a constraint that explains under what conditions the course can be approved.

- **Derivations**: Derivations enable the organisation to know new knowledge from existing knowledge through explicitly defined algorithms. For example, *cost of a course is calculated by adding staff costs, facilities costs, overheads, and government taxes*.

The role of these four types rules in defining process elements and their associations are depicted in Figure 5.22.
There are five types of derivations (numbered from (1) to (5) in Figure 5.22) carried out in different stages, based on different types of rules. Firstly, all four types of rules are derived or identified (labelled as (1) in Figure 5.22) based on the contextual artefacts (Hay et al., 1997). Then term definition rules are used to derive the process elements (labelled as (2) in Figure 5.22). Thirdly, facts relating rules would assist in defining the three types of process element associations (labelled as (3) in Figure 5.22). These are associations of actions with participants- \( f(A,P) \), associations of actions with data elements- \( f(A,D) \) and associations among data components that create integrity relationships- \( f(D) \). Fourthly, assertion or constraining rules are used to define the associations between process actions- \( f(A) \) and all types of constrictive associations \( C\{f(A,P)\}, \ C\{f(A)\}, \ C\{f(A,D)\} \) and \( C\{f(D)\} \). Finally, the computational associations between data elements- \( f(D) \) are defined based on derivation rules.

When elements and their associations are introduced to a process, there is always a rationale for it. These rationales were previously introduced in Chapter-1 as:

- **Value adding reasons** – This refers to any process element and/or associations introduced to contribute to the end goal of the process.
• **Gate-keeping reasons** – Indicates any process element and/or associations introduced for control purposes such as security, quality and managerial.

• **Communication purposes** – This refers to any process element and/or associations introduced only for the task of informing other personal, processes, and/or organisations that are not directly linked with the process.

Some process elements and their associations may satisfy more than one rationale listed above. For example, in OCAS processes, when resources are checked by resource divisions, it serves both gate-keeping and communication purposes.

Based on the above-discussed aspects, the schema that captures the associations between elements in the **pragmatic and semantic levels** requires the following features:

- Ability to refer to every process elements and its association with a contextual artefact (or a fragment of it).
- Ability to define the type of derivation carried out.
- Ability to define more than one rationale for the association.

Figure 5.23 presents the schema proposed to capture **inter-level correlations** of elements between the **pragmatic and semantic levels**.
The schema proposed in Figure 5.23 has six entities. Out of these six entities, two (contextInfoRegistry and fragmentRegistry) are from the previous model that captured CAD in the context level. There are four entities newly introduced to this model:

- **Process Element Registry** (*ProcessElementRegistry*) – Previously in section 5.2.2 only a graphical model was introduced to capture the process elements and its associations. That discussion also mentioned that a formalism would be introduced in the next chapter, to capture these associations. In order to link the elements used in the formalism with the rest of the model, they are required to be stored in a registry. This entity (*ProcessElementRegistry*) serves that purposes of listing all the process elements and associations in a flat file.

- **Rationales** – Records a set of rationales (value adding, gate keeping and communication purposes) in this entity for referencing.

- **Context and Process Element Association** (*ContextProcessAssociation*) – This associative entity captures correlations that exist between contextual information artefacts (or fragments of it) and process elements. As mentioned in the note in Figure 5.23, attribute *TypeOfDerivation* allows capturing the appropriate derivation method (discussed earlier) according to the process element and contextual artefact.
• **Rationale for Process Element** (*RationaleForProcessElement*) – This associative entity records many-to-many relationship between entries in the ContextProcessAssociation entity and rationales.

This model (Figure 5.23) facilitates capturing associations between process elements and contextual artefacts (or fragments of it). In addition, the types of derivations and rationale for the process elements’ (or its associations) existence, can be recorded in this model. This model, similar to other schemata (Figures 5.2, 5.20 and 5.21) introduced before, contributes to the **PECIA Model** in Chapter-8.

### 5.3.2 CAD between Elements in the Semantic and Syntactic Levels

The process elements (actions, participants, and business object) are generally abstracted in models (data, process, and interface) prior to implementation. These models are the elements in the *syntactic level*. Process elements or the associations among them, can be modelled more than in one syntactic model. For example in OCAS, a business object is captured in an object-relational model and referenced in UI definition model.

The requirement here is for a simple schema that facilitates capturing the many-to-many relationships that exist between process elements (and its associations) and models in the *syntactic level*. The schema presented in Figure 5.24 is proposed for recording the CAD between elements in the *semantic* and *syntactic levels*.

In this schema, ProcessModelAssociation is the only new entity. The two entities ProcessElementRegistry and ModelRegistry were previously described in schemata presented in Figures 5.23 and 5.20 respectively.
Chapter 5-Constraints, Associations, and Dependencies (CAD) of Process Elements

5.3.3 CAD between Elements in the Syntactic and Implementation Levels

The reason for creating models (in the syntactic level) is to facilitate the implementation. Clearly, this shows a many-to-many associations that exist between models (data, process and interface) and implementation artefacts (databases, code files, documents and scripts).

The schema given in Figure 5.25 is proposed to capture the above-mentioned many-to-many relationships that exist between elements in the syntactic and implementation levels. This schema is similar to the previous schema (Figure 5.24) that captures the associations of elements in the semantic and syntactic levels. This schema (in Figure 5.25) has two entities previously introduced in models given in Figure 5.21 and 5.23. The new entity in this model is ModelWebArtefactAssociation, which is an associative entity.

The entity named ProcessModelAssociation (process and model association) allows recording the many-to-many relationships between process elements and models represented in the syntactic level. This model also contributes to the PECIA Model presented later in this thesis.
Chapter 5: Constraints, Associations, and Dependencies (CAD) of Process Elements

5.4 Chapter Summary

This chapter presents a study of CAD (Constraints, Associations and Dependencies) among elements in the four levels (pragmatic, semantic, syntactic, and implementation) in the PoPA framework. The correlations within and across these levels are studied under two topics: intra-level CAD and inter-level CAD.

The main contributions of this chapter (to the end goal of this research) are six schemata that facilitate capturing intra and inter-level CAD. These schemata are developed based on findings taken from the main case study (OCAS), the literature and author’s experience in web-application development and process modelling. These schemata can capture the correlations listed below:

- **intra-level CAD** among elements in the pragmatic level (Figure 5.2)
- **intra-level CAD** among elements in the syntactic level (Figure 5.20)
• *intra-level CAD* among elements in the *implementation level* (Figure 5.21)

• *inter-level CAD* between the *pragmatic* and *semantic levels* (Figure 5.23)

• *inter-level CAD* between the *semantic* and *syntactic levels* (Figure 5.24)

• *inter-level CAD* between the *syntactic* and *implementation levels* (Figure 5.25)

All of the above-developed schemata contribute to the **PECIA Model** presented later in Chapter-8 of this thesis.

This chapter did not present a specific schema to capture CAD in the *semantic level*. However, an extensive study was performed to comprehend the CAD among process elements (in the *semantic level*). The outcome of this study was a diagrammatic representation of CAD among process elements (Figure 5.18). Due the complex nature of the CAD in the *semantic level*, it requires a suitable formalism rather than a simple relational schema. Therefore, the search for a proper formalism and representation of CAD among process elements using that formalism will be presented in the next chapter.
Chapter-6

“Mathematics is the language with which God has written the universe”
Galileo Galilei (1564 - 1642)

“The art of reasoning can be reduced to a well-constructed algebraic expression”
Antoine Lavoisier (1743-1794)

6 KAT for Conceptualising CAD among Process Elements

6.1 Chapter Overview

Business process elements - actions, participants and object attributes, are interwoven together to achieve business goals. To attain this interconnection, there are a number of constraints, associations and dependencies (CAD) imposed among these process elements. The main objective of this chapter is to find a suitable formalism to model CAD among process elements.

Previously Chapter-5 detailed the nature of correlations that exist among process elements, under the heading ‘intra-level CAD within elements in the semantic level’ (Semantic level, in the paradigm of process automation– PoPA framework introduced in Chapter-1, represents the operational business process). The discussion in Chapter-5 presented CAD among process elements in a diagrammatic form - Model of CAD among Process Elements (Figure 5.18 of Chapter-5).

Model of CAD among process elements manifested the correlations among process elements into eight forms as follows:

- **Associations and Dependencies within Process Actions** - f(A): This refers to different types of associations - Sequential, Conditional Split, Parallel Split, Simple Merge, Multi Merge, Iteration, and Skip, that can exist within process actions.
• **Associations and Dependencies between Process Actions and Participants** – \( f(A,P) \): This denotes three types of associations - obligations, permissions, and forbiddances, that exist between process actions and participants.

• **Associations and Dependencies between Process Actions and Business Object** – \( f(A,D) \): This signifies Data Visibility in process actions, Data Interaction between process actions, and Data-based routing of process actions, as the three types of associations between actions and business object (data).

• **Associations and Dependencies within Business Object** – \( f(D) \): This indicates integrity and computational relationships among business process data elements.

• **Constraints on the associations within Process Actions** – \( C\{f(A)\} \): This implies the constraints; such as work item failures, time constraints, participant or data unavailability, external triggers, and other constraints triggers, that can be imposed on the previously mentioned associations among process actions - \( f(A) \).

• **Constraints on the associations between Process Actions and Participants** – \( C\{f(A,P)\} \): Three categories of constraints are identified to affect the associations between actions and participants as: i) characteristics possessed by individual participants, ii) types of relationships between two participants and iii) external factors.

  o **Individual characteristics** include Location, Experience, Skills or Capabilities, Availability, Workload, History of performing process actions previously, Preference, Belonging to an organisational sub-structure and Level of the organisational role of the participant.

  o **Relationships between Participants** refer to factors such as Belonging to same or different sub-structures, Reporting and Delegation Authorities, and Inter personal relationships of two participants.

  o **External factors** include Time constraints and Method of allocation of actions to participants deployed by the workflow management system.

• **Constraints on the associations between process Actions and Business Object** - \( C\{f(A,D)\} \): This represents two constraints ownership of data by an individual and specialisation of business object that defines the association between actions and data.
• **Constraints on the associations within Business Object Elements - C(f(D))**: This implies *specialisation* constraint that could affect the previously identified integrity and computational relationships.

The modelling of above CAD among process elements in a suitable formalism would facilitate locating the propagating impact of process element changes. As the correlations among process elements are complex it is challenging to model them cohesively in an object-relational data structure. Therefore, many (Camara et al., 2005; van der Aalst, 1997; Fokkink et al., 1994; Murata, 1989) have used formalisms. In addition, a suitable formalism that captures these complex correlations among process elements into linear expressions would facilitate the level of analysis required by this research. This chapter, therefore initially searches for an appropriate formalism that can model all types of CAD among process elements, into a linear expression. Further, binary tree representations are used to abstract these linear expressions for implementation purposes.

The possible candidate formalisms are Petri-Nets, Process Algebra, and Kleene Algebra with Tests (KAT). Petri-Nets is a popular visual process-modelling tool, which is backed up by sound mathematical algorithms (van der Aalst, van Hee, & Houben, 1994; Murata, 1989; Reisig, 1985). Process Algebra allows processes to be captured into analytical linear expressions, which are suitable for computer manipulation (Basten, 1998; Fokkink et al., 1994; van Glabbeek, 1987). Kleene Algebra with Tests (referred as KAT) (Kozen, 1999) is a bi-partite algebra that has extended itself for various other modelling purposes such as web applications (Schewe et al., 2005). However, currently there is no known application of KAT for process modelling. These three formalisms are closely analysed to gauge their suitability to capture the CAD among process elements.

This chapter is organised as follows. First a suitable formalism is selected between - Petri-Nets, Process Algebra, and KAT. Then the features of this selected formalism are closely analysed to comprehend its proper usage. Followed by this study of the formalism, each of the above-listed CAD among process elements is systematically modelled, to create an analytical linear expression of a process. In this discussion, the use of the selected formalism to model CAD among process elements is exemplified using a sample process drawn from the case study (Online Course Approval System - OCAS was detailed as the case study of this research in Chapter-
4). The final section of this chapter demonstrates a mechanism to organise linear process expressions into a binary tree structure, which can be algorithmically analysed. These binary-tree based process expressions contribute to the **PECIA (Process Evolutions and Change Impact Analysis) Model** (presented in Chapter-8) as the core solution the problem of effectively managing business process evolutions and changes in web workflows.

### 6.2 Selection of a Suitable Formalism

There are a number of formal process-modelling tools to capture the inner workings of a process. The literature review carried out in Chapter-3 identified Petri-Nets (van der Aalst et al., 1994; Murata, 1989; Reisig, 1985), Process Algebra (Basten, 1998; Fokkink et al., 1994; van Glabbeek, 1987) and Kleene Algebra with Tests –KAT (Kozen, 1999; Kozen, 1997) as possible candidate formalisms that can serve the modelling requirements of this research. This section selects the most suitable formalism among Petri-Nets, Process Algebra and KAT, to model the complex correlations that exist among process elements.

Prior to the selection of a formalism, the following factors are identified as the requirements that need to be met by the selected formalism, in modelling CAD among process elements:

- Ability to represent the eight types of CAD among process elements identified in Chapter-5, and listed in the overview of this chapter, in the following two categories:
  - **Dynamic associations** that show the correlations among actions—Sequence, Parallel, Conditional Choice, Synchronisation, and simple merge. These show the dynamic nature of a process and are identified as the first type of CAD among process elements in the above-mentioned list.
  - **Static associations** classify the other seven types of associations; within process participant and object, and between actions, participants, and object.

- Ability to capture both dynamic and static associations among process elements, into a single linear expression that allows computer manipulation.
Table 6.1 presents a comparison between Petri-Nets, Process Algebra, and KAT based on their ability to achieve the afore-mentioned requirements. Following symbols are used to indicate the degree of support; (+) indicates that the formalism fully supports the requirements without any modifications, (-) indicates the inability to support the requirement and (>) indicate the ability to support with certain known extensions or minor modifications.
Table 6.1- Comparison of the ability of three different Formalisms in capturing CAD among Process Elements

<table>
<thead>
<tr>
<th>Formalism</th>
<th>Brief Description / Axioms</th>
<th>Dynamic Associations</th>
<th>Static Associations</th>
<th>Create Linear expressions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Petri-Nets</td>
<td>Petri-Nets is a graphical modelling tool that consists two types of elements (bi-partite): places and transitions. Flow controls are presented in graph form, while actions are presented in transitions and action post and pre-conditions are presented in places. (van der Aalst et al., 2003b)</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Process Algebra</td>
<td>Process Algebra is a uni-partite algebraic structure (A, +, •,</td>
<td></td>
<td>), where +, • and</td>
<td></td>
</tr>
<tr>
<td>Kleene Algebra with Tests (KAT)</td>
<td>KAT is a two-sorted (bi-partite) algebraic structure (B, K, +, •, *, 0, 1, ~), where B is a subset in K. ~ is a unary operator (similar to negation) defined only on B. * is also a unary operator is similar to iteration defined only on elements from K. 0 and 1 are special elements that symbolise null and skip actions. (Kozen, 1997)</td>
<td>+</td>
<td>&gt;</td>
<td>+</td>
</tr>
</tbody>
</table>
Although Petri-Nets based models are appropriate for creating graphical process notations, according to the findings presented in Table 6.1, in Petri-Nets it is not able to create linear process expressions as required by this research. However, Process Algebra can create linear expressions of processes, but unable to capture certain synchronisations and static associations in an efficient manner, due to its uni-partite (single type of element) nature (van der Aalst, 2003). Identifying this gap, researchers such as Hall (1993) and Mikolajczak (2002) combines Petri-Nets with Process Algebra into a Calculi named Petri-Nets Calculus or Petri Box Calculus (PBC). Still there are certain issues with PBC as it is not as expressive when compared to KAT.

KAT being a uni-partite algebra is able to capture both dynamic and static associations. Further, it is possible to create linear expressions that can be analysed as required by this research. In addition, KAT has special elements such as 0 and 1 that represents null and skip actions respectively. PBC lacks such null or skip actions representations and other unary operators such as iterations (*) and negations (~). In this view, KAT stands out to be the most suitable formalism to capture CAD among process elements.

One of the disadvantages of KAT over Process Algebra (and PBC) is that it lacks a direct operator to present parallels. However, using the choice (+) and iteration (*) operators in combination, the parallel flow control can be constructed in KAT expressions. The use of choice and iteration in combination to represent parallel flow constructs is presented later in this chapter.

### 6.3 Kleene Algebra with Tests (KAT)

Kleene Algebra was first introduced by Stephen C. Kleene (1956). Initial Kleene Algebra, was a uni-partite algebra, which had only a single element. Forty years after the introduction of Kleene Algebra, Kozen (1997) introduced an extension named Kleene Algebra with Tests (KAT). As opposed to the original Kleene Algebra, KAT is a bi-partite algebra (two types of elements) (Kozen, 1999).

The axioms of KAT are as follows:
- KAT is a two-sorted (bi-partite) algebraic structure (B, K, +, •, *, 0, 1, ~), where B is a subset in K
• + is commutative such that \( a + b = b + a \) for all \( a, b \) in \( K \) and \( B \)
• \( \cdot \) is commutative such that \( a \cdot b = b \cdot a \) for all \( a, b \) only in \( B \)
• + and \( \cdot \) are distributive such that \( a(b + c) = (ab) + (ac) \) and \( (b + c)a = (ba) + (ca) \) for all \( a, b, c \) in \( K \) and \( B \)
• for + and \( \cdot \) there exists an element 0 in \( K \) and \( B \) such that for all \( a \) in \( K \): \( a + 0 = 0 + a = a \) and \( a0 = 0a = 0 \)
• for + and \( \cdot \) there exists an element 1 in \( K \) and \( B \) such that for all \( a \) in \( K \): \( a1 = 1a = a \)
• for \( * \) there exists an elements 1 and a only in \( K \) such \( 1+aa* = a \) and \( 1+ a*a = a \).
In other words \( * \) behaves like the Kleene Star operator in formal language theory (Kozen, 1997). This is similar to iteration in process elements.
• and \( \sim \) is a unary operator, similar to negation, defined only on \( B \)

According to above axioms \((K, +, \cdot, *, 0, 1)\) is a Kleene algebra and \((B, +, \cdot, \sim, 0, 1)\) is a Boolean algebra (Kozen, 1997).

Initially KAT was sparsely used for practical modelling purposes. However, with the interest in web engineering domain to conceptualise web applications, Schewe and Thalheim (2005) demonstrated a practical use of KAT to model web applications. Their use of KAT inspired the author to experiment with KAT for modelling business processes in this research.

When modelling processes there are two types of associations – dynamic and static that needs to be captured. Dynamic associations relate to process actions or events and static associations relates to data or conditions. In Petri-Nets, \textit{transitions} are used to capture process actions and \textit{places} are used to capture pre and post conditions or input data (Adam et al., 1998; van der Aalst, 1997; Murata, 1989).

In KAT, the elements in \( K \) are similar to transitions in Petri-Nets (Mikolajczak, 2002; Murata, 1989), thus can represent events, computation step, signal processor, task or job and clause in logic (Adam et al., 1998; van der Aalst, 1997; Murata, 1989). Elements from \( B \) (\( B \) is a sub set of \( K \)) are called tests, conditions or guard elements. These are similar to places defined under Petri-Nets (Mikolajczak, 2002; Murata, 1989). Thus can hold, process related pre and post conditions, input data, input signals and process resources or participants (Adam et al., 1998; van der Aalst, 1997).
In the context of this thesis, two terms are introduced to refer to elements in K and B as **alphas** and **phis** respectively.

Elements of K are denoted by $\alpha_1, \alpha_2, \ldots, \alpha_n$ and referred to as **alphas** (see Definition-1). Alphas represent process actions, since they are similar to transitions in Petri-Nets.

**Definition-1:** Actions of a process is denoted by $a \Rightarrow \{\alpha_1, \alpha_2, \ldots, \alpha_i\}$, which is a subset of K with an algebraic structure $(K, a, +, \cdot, *, 0, 1)$

Elements from B are symbolised by $\phi_1, \phi_2, \ldots, \phi_m$ and referred to as **phis** (see Definition-2). Phis denote constraints or guard elements, as they are similar to place holders in Petri-Nets.

**Definition-2:** Conditions or guard elements acting on actions are denoted by $c \Rightarrow \{\phi_1, \phi_2, \ldots, \phi_j\}$ is a subset of B with an algebraic structure $(B, c, +, \cdot, ~, 0, 1)$

Using KAT axioms the two types of elements - **alphas** and **phis** can be presented in a single expression (Schewe et al., 2005; Kozen, 1997; Koch et al., 1996). The following Definition-3 introduces the basic rule in presenting these two types of elements in a single expression.

**Definition-3:** If $\alpha_x \in a$ represents an action where $(K, a, +, \cdot, *, 0, 1)$ and $\phi_y \in c$ denotes a condition where $(B, c, +, \cdot, ~, 0, 1)$; constraint imposed on action is represented using the expression $\phi_y \alpha_x$. This implies that action $\alpha_x$ can be performed iff constraint $\phi_y$ is satisfied.

Generally, in writing KAT expressions, the dot operator is omitted for readability (Kozen, 1999).

Founded on the above Definitions-1, 2 and 3 the next section introduces a series of definitions to capture different types of correlations among process elements.
6.4 KAT for Modelling Business Processes

This section systematically develops a series of KAT definitions to model CAD among process elements. Sections will unfold in eight sub-headings as: i) Associations among Actions, ii) Associations between Actions and Participants, iii) Associations between Actions and Object Data, iv) Associations among Object Data, v) Constraints on the associations among Actions, vi) Constraints on the associations between Actions and Participants, vii) Constraints on the associations between Actions and Object, and viii) Constraints on the associations within Object Data.

Under each sub-heading, one or more KAT based definitions are introduced. In addition, practical use of these definitions are exemplified within each sub-heading, using the New Course Approval Process automated in OCAS Project (OCAS project was detailed in Chapter-4).

6.4.1 Representing Associations among Actions using KAT

Associations among actions refer to types flow controls identified as sequential, conditional split, parallel split, simple merge, multi merge, iteration and skip. Here KAT based definitions will be presented to capture each of these flow controls.

Consider the two actions denoted using $\alpha_x$ and $\alpha_y$. Between these two actions, if $\alpha_x$ requires to be completed before the action $\alpha_y$ (Figure 6.1).

![Figure 6.1- Sequential Composition between two Actions](image)

The Definition-4 gives a construct to represent the sequential composition between two actions.

**Definition-4**: The sequential association between two action $\alpha_x, \alpha_y$, where $(K, a, +, *, 0, 1)$, is represented as $\alpha_x \alpha_y$. The expression $\alpha_x \alpha_y$ denotes that $\alpha_x$ followed by $\alpha_y$. Here $\alpha_x \alpha_y \neq \alpha_y \alpha_x$ due to the non-commutative nature of the dot operator.
Consider the two actions denoted using $a_x$ and $a_y$. Between these two actions, either $a_x$ or $a_y$ can take place (Figure 6.2).

![Figure 6.2- Conditional Choice between two Actions](image)

This representation of choice between two actions is given in Definition-5 below.

**Definition-5:** The choice between two actions $a_x, a_y \in a$, where $(K, a, +, \cdot, *, 0, 1)$, is represented as $(\phi_x a_x + \phi_y a_y)$ or $(\phi_y a_y + \phi_x a_x)$. In this $\phi_x$ and $\phi_y$ are guard element such that $\phi_x, \phi_y \in c$ and $(B, c, +, \cdot, ~, 0, 1)$. These constructs $(\phi_x a_x + \phi_y a_y)$ or $(\phi_y a_y + \phi_x a_x)$ denote either action $a_x$ or $a_y$, can take place based on the true guard condition $\phi_x$ or $\phi_y$, respectively. Here $(\phi_x a_x + \phi_y a_y) = (\phi_y a_y + \phi_x a_x)$ due to the + operator’s commutative nature.

In a process, certain actions would be required to iterate, until it satisfies a particular condition (see Figure 6.3). For example, in a course approval process, the action creating unit information is repeated, until all unit details of the course are entered into the system.

![Figure 6.3- Looping or Iteration of an Action until a Guard Condition is satisfied](image)

The Definition-6 presents the construct for iteration of actions.

**Definition-6:** The iteration of an action $a_x \in a$, where $(K, a, +, \cdot, *, 0, 1)$, is represented as $(\phi_i a_x)^*$, in which $\phi_i \in c$, where $(B, c, +, \cdot, ~, 0, 1)$. The construct $(\phi_i a_x)^*$ indicate that action $a_x$ can be repeated until the looping condition $\phi_i$ is satisfied.
Consider the scenario, where the two actions $\alpha_x$ and $\alpha_y$ needs to take place in parallel (see Figure 6.4).

\[ \phi_m(\phi_x \alpha_x + \phi_y \alpha_y)^* \]

\text{Figure 6.4 - Parallelism between two Actions}

KAT does not provide a construct to represent parallelism directly as in Process Algebra. Schewe and Thalheim (2005) argue that parallelism is a differed choice, given the opportunity to repeat the choosing. However, a particular action should be chosen once and only once, in each iteration. The selection of an action can be constrained by appropriate guard conditions. This construct is demonstrated in Definition-7.

**Definition-7:** The parallelism between two action $\alpha_x, \alpha_y \in a$, where $(K, a, +, \cdot, *, 0, 1)$, is represented as $(\phi_m(\phi_x \alpha_x + \phi_y \alpha_y))^*$. In this $\phi_x, \phi_y$ and $\phi_m$ are guard elements such that $\phi_x, \phi_y, \phi_m \in c$ and $(B, c, +, \cdot, \sim, 0, 1)$. This construct $(\phi_m(\phi_x \alpha_x + \phi_y \alpha_y))^*$ indicate that choice between actions $\alpha_x$ and $\alpha_y$ can be looped, until the merging condition $\phi_m$ is satisfied. In addition, the choice of action $\alpha_x$ or $\alpha_y$ is guarded by $\phi_x$ and $\phi_y$, allowing the actions to be chosen only once.

In certain process instances, some actions may be skipped, without affecting the main flow of the process. For example, in *online money transfer processes*, checking of exchange rates by a participant may not required to be performed for some instances and this would not affect the normal flow of the process. This aspect is particularly applicable to web-based workflows, where additional information could be easily linked to action interfaces, allowing the user to view additional information such as help tips, FAQs, terms and conditions and any other relevant policy document. Definition-8 shows the mechanism to represent skipping of an action without affecting the main flow of the process.
Definition-8: If the process action $a_x 1ca$, where $(K, a, +, *, 0, 1)$, it can be denoted as $(a_x + 1)$. This construct means that either action can be performed or skipped.

6.4.1.1 Application of KAT Definitions-1 to 8 in OCAS

To exemplify the use of above Definitions-1 to 8, OCAS related running example of the New Course Approval Process is used.

In the New Course Approval Process, there are over thirty actions, listed in Table 6.2 below using a set of alpha notations according to the Definition-1.

Table 6.2- Alphas related to the OCAS New Course Approval Process

<table>
<thead>
<tr>
<th>Reference no in Figure 6.5</th>
<th>Alpha notations</th>
<th>Action Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$a_1$</td>
<td>Create Course Notification of Intention (CNI)</td>
</tr>
<tr>
<td>2</td>
<td>$a_2$</td>
<td>Submit CNI</td>
</tr>
<tr>
<td>3</td>
<td>$a_3$</td>
<td>Recommend CNI</td>
</tr>
<tr>
<td>4</td>
<td>$a_4$</td>
<td>Approve CNI</td>
</tr>
<tr>
<td>5</td>
<td>$a_5$</td>
<td>Create Full Course Proposal (FCP)</td>
</tr>
<tr>
<td>6</td>
<td>$a_6$</td>
<td>Create unit information</td>
</tr>
<tr>
<td>7</td>
<td>$a_7$</td>
<td>Submit FCP</td>
</tr>
<tr>
<td>8</td>
<td>$a_8$</td>
<td>Assess academic quality and recommend FCP</td>
</tr>
<tr>
<td>9</td>
<td>$a_9$</td>
<td>College Recommendation for FCP</td>
</tr>
<tr>
<td>10</td>
<td>$a_{10}$</td>
<td>Course Approvals Committee Recommendation for FCP</td>
</tr>
<tr>
<td>11</td>
<td>$a_{11}$</td>
<td>Approve FCP</td>
</tr>
<tr>
<td>12</td>
<td>$a_{12}$</td>
<td>Fill in the course code information</td>
</tr>
<tr>
<td>13</td>
<td>$a_{13}$</td>
<td>Commence internal forum consultation</td>
</tr>
<tr>
<td>14</td>
<td>$a_{14}$</td>
<td>Submit internal forum consultation report</td>
</tr>
<tr>
<td>15</td>
<td>$a_{15}$</td>
<td>Commence external advisory committee consultation</td>
</tr>
<tr>
<td>16</td>
<td>$a_{16}$</td>
<td>Submit external advisory committee consultation report</td>
</tr>
<tr>
<td>17</td>
<td>$a_{17}$</td>
<td>Create and submit Resources Assessment (RA) document</td>
</tr>
<tr>
<td>18</td>
<td>$a_{18}$</td>
<td>Office of Marketing Sign-off on the RA document</td>
</tr>
<tr>
<td>19</td>
<td>$a_{19}$</td>
<td>IT directorate’s (ITD) sign-off on the RA document</td>
</tr>
<tr>
<td>20</td>
<td>$a_{20}$</td>
<td>Education Development Unit’s (EDU) sign-off on the RA document</td>
</tr>
<tr>
<td>21</td>
<td>$a_{21}$</td>
<td>Library sign-off on the RA document</td>
</tr>
<tr>
<td>22</td>
<td>$a_{22}$</td>
<td>Office of Academic Registrar–System’s (OAR-Sys) sign-off on RA document</td>
</tr>
<tr>
<td>23</td>
<td>$a_{23}$</td>
<td>Office of Academic Registrar–Operation’s (OAR-opr) sign-off on RA document</td>
</tr>
<tr>
<td>24</td>
<td>$a_{24}$</td>
<td>Timetabling sign-off on the RA document</td>
</tr>
<tr>
<td>25</td>
<td>$a_{25}$</td>
<td>Capital Works sign-off on the RA document</td>
</tr>
<tr>
<td>26</td>
<td>$a_{26}$</td>
<td>Approve RA document</td>
</tr>
<tr>
<td>27</td>
<td>$a_{27}$</td>
<td>Create Business Case (BC) document</td>
</tr>
<tr>
<td>28</td>
<td>$a_{28}$</td>
<td>Assess and Sign-off the BC with an independent statement</td>
</tr>
<tr>
<td>29</td>
<td>$a_{29}$</td>
<td>Recommend BC</td>
</tr>
<tr>
<td>30</td>
<td>$a_{30}$</td>
<td>Vice Chancellor’s Advisory Committee Approval for BC</td>
</tr>
<tr>
<td>31</td>
<td>$a_{31}$</td>
<td>Business and Finance Committee Approval for BC</td>
</tr>
<tr>
<td>32</td>
<td>$a_{32}$</td>
<td>College Approval for BC</td>
</tr>
<tr>
<td>33</td>
<td>$a_{33}$</td>
<td>View course details</td>
</tr>
</tbody>
</table>
Figure 6.5 - Actions grouped in to Sub-Processes in the New Course Approval process of OCAS
The actions listed in Table 6.2 can be grouped into seven sub-processes (SP1 – SP7) as in Figure 6.5. These actions are given alpha notations from $\alpha_1$ to $\alpha_{33}$ (in Table 6.2).

Using these $\alpha_1$ to $\alpha_{33}$ notations and Definitions-1 to 8, each sub-processes in Figure 6.5, is given a KAT expressions (see following KAT Expressions-6.1 to 6.7). In these KAT expressions, there are a number of guard elements identified from $C_1$ to $C_{32}$, to signify relevant constraints. In these high-level KAT expressions as $C_1$ to $C_{32}$ are as composite conditions, represented as placeholders, which will be detailed later with the help of other definitions that are yet to be introduced in this chapter.

**SP1:= $C_1\alpha_1C_2\alpha_2C_3\alpha_3C_4\alpha_4$**

*KAT Expression 6.1- Sub Process of CNI creation in New course approval process*

**SP2:= $C_5\alpha_5 C_6\alpha_6*$ $C_7\alpha_7C_8\alpha_8C_9\alpha_9C_{10}\alpha_{10}C_{11}\alpha_{11}$**

*KAT Expression 6.2- Process of FCP creation in New course approval process*

**SP3:= $C_{12}\alpha_{12}$**

*KAT Expression 6.3- Sub Process of course code entry in New course approval process*

**SP4:= $C_{13}\alpha_{13}$ $C_{14}\alpha_{14}$**

*KAT Expression 6.4- Sub Process of internal forum report in New course approval process*

**SP5:= $C_{15}\alpha_{15}$ $C_{16}\alpha_{16}$**

*KAT Expression 6.5- Sub Process of external advisory committee report in New course approval process*

**SP6:= $C_{17}\alpha_{17}$ $C_n(C_{18}\alpha_{18} + C_{19}\alpha_{19} + C_{20}\alpha_{20} + C_{21}\alpha_{21} + C_{22}\alpha_{22} + C_{23}\alpha_{23} + C_{24}\alpha_{24} +$ $(C_{25}\alpha_{25} + \sim C_{25} (\alpha_{25} +1)))$ $C_{26}\alpha_{26}$**

*KAT Expression 6.6- Sub Process of Resource Assessment in New course approval process*

**SP7:= $C_{27}\alpha_{27}$ $C_{28}\alpha_{28}$ $C_{29}\alpha_{29}$ $(C_{30}\alpha_{30} + C_{31}\alpha_{31} + C_{32}\alpha_{32})$**

*KAT Expression 6.7- Sub Process of Business Case Approval in New course approval process*

- **KAT Expression 6.1:** This denotes series of actions, associated with *course notification of intention-CNI* sub process, that take place in sequence as depicted in label $SP1$ in Figure 6.5.
• **KAT Expression 6.2:** This refers to the sub process \( SP2 \) denoted in Figure 6.5. This sub process is associated with approval of *full course proposal-FCP*. Most of the actions in this process take place in sequence, except for the action *Create unit information-\( a_6 \)*. This action is repeated until all units are entered. In this, the merging condition \( C_6 \) assures that each individual unit details are entered once and only once.

• **KAT Expression 6.3:** This is the simplest sub-process, *entry of course code information* that consists of only one action.

• **KAT Expression 6.4 and 6.5:** These two expressions present the *SP4-internal forum* and *SP5-external advisory committee consultation* sub-processes. Structure wise, these expressions are also relatively simple, presenting a set of actions that take place in sequence, under appropriate guard conditions.

• **KAT Expression 6.6:** This sub process refers to the *SP6-resource assessment process*. The intricate part of the structure is the resource divisional sign-offs that can take place in parallel. According to Definition-6, this is presented using the construct \( C_n(C_{18}a_{18} + \ldots + C_{25}a_{25}) \). In this construct the \( C_n \) represents the merging condition, that checks whether all the sign-offs are given. Also in this construct, it shows a possible skipping of the actions \( a_{25} - Capital\ Works\ sign-off\ on\ the\ RA\ document\). This is presented as \( (C_{25}a_{25} + \sim C_{25}(a_{25} +1)) \). This indicates if the composite guard condition \( C_{25} \) is true, the strict necessity to perform the action \( a_{25} \). If the guard condition is not true (\( \sim C_{25} \)) it is optional to perform the action \( a_{25} \) or could be skipped altogether.

• **KAT Expression 6.7:** This expression represents the sub-process *SP7-business case approval*, in the *New Course Approval Process*. The early actions of this process are sequential, but the last action is a conditional choice.

When considering the *New Course Approval Process* as a whole, the seven notations SP1 to SP7 can be considered as composite actions. This indicate that SP1, SP2, ..., SP7 are elements from K and could be included in a KAT expression as
alphas. Thus, all the definitions from Definition-1 to 8 are applicable. Therefore, by combining the KAT Expressions-6.1 to 6.7, a composite KAT Expression E can be written as follows;

\[
E := c_{sp1} c_{m}(c_{sp2} + c_{sp3} + c_{sp4} + c_{sp5} + c_{sp6} c_{sp7})^*
\]

**KAT Expression 6.8- Composite statement of the New Course Approval Process**

This KAT Expression 6.8 can be expanded using the previous expressions from 6.1 to 6.7. The full expression for the *New Course Approval Process* is given in KAT Expression 6.9.

\[
E := C_{1} a_{1} C_{2} a_{2} C_{3} a_{3} C_{4} a_{4} C_{m} ( (C_{15} a_{15} C_{16} a_{16}) + (C_{5} a_{5} C_{6} a_{6})^* \\
C_{7} a_{7} C_{8} a_{8} C_{9} a_{9} C_{10} a_{10} C_{11} a_{11}) + (C_{12} a_{12} + C_{13} a_{13} C_{14} a_{14}) + (C_{15} a_{15} C_{16} a_{16}) + (C_{17} a_{17} C_{n}(C_{18} a_{18} + C_{19} a_{19} + C_{20} a_{20} + C_{21} a_{21} + C_{22} a_{22} + C_{23} a_{23} + C_{24} a_{24} + (C_{25} a_{25} + ~C_{25} (a_{25} +1)))^* C_{26} a_{26} (C_{27} a_{27} C_{28} a_{28} C_{29} a_{29} (C_{30} a_{30} + C_{31} a_{31} + C_{32} a_{32})))^*
\]

**KAT Expression 6.9- Expanded New course approval process, using all action notations**

This expression captures all process elements and dependencies that exist among those elements. However in this expression other associations and dependencies are presented using composite elements \(C_1\) to \(C_{32}\), \(C_m\) and \(C_n\), which are gradually developed in later sections.

### 6.4.2 Representing Associations between Actions and Participants using KAT

The associations between process actions and participants were previously identified to be *obligations, permissions, and forbiddances*. Schewe and Thalheim (2005) have presented a mechanism for modelling web applications using KAT. In this research, the author uses the same technique to show the association between process actions and participants. Definitions-10, 11 and 12 below are adapted from the work of Schewe and Thalheim (2005).

Permission to carry out an action is defined as follows;

**Definition-9:** \(\phi_x \rightarrow P \text{ do (role, action)}\)

In this definition letter \(P\) indicate the *permission* trait. Specific role of the participant and action is associated within brackets.
Similar to the above Definition-9, the obligation to carry out action is presented in Definition-10.

**Definition-10:** $\phi \rightarrow O$ do (role, action)

The **Forbiddance** or prohibition to carry out an action is given in Definition-11.

**Definition-11:** $\phi \rightarrow F$ do (role, action)

### 6.4.2.1 Application of KAT Definitions-9, 10 and 12 in OCAS

In the **New Course Approval Process** (Figure 6.5), there are participants associated with each of the 33 actions. In some cases, the same action is associated with more than one participant. For example in **full course approval process** (SP2), the action $\alpha_{10}$-Course Approvals Committee Recommendation for FCP is required to be performed by the **Course Approval and Articulation Committee** (CAAC). Since this is a committee decision, the CAAC **committee chair** has the obligation to perform this action. However, in reality the committee decision can be also actioned by the **secretary to the CAAC**. The secretary to the CAAC has the permission, but not an obligation to perform the action on behalf of the chair.

Table 6.3 below presents the associations that exist between process participants and actions of the **New Course Approval Process**, based on Definitions-9, 10 and 11. Here a set of Phis ($\phi_1$ to $\phi_{37}$) are introduced. Section 6.2 previously explained that phis are elements from the space $B \in K$ according to the KAT axioms.

**Table 6.3- Association between Process Participants and Actions in the New Course Approval Process**

<table>
<thead>
<tr>
<th>KAT representation to present schema level constraint</th>
<th>Role of the person</th>
<th>Action reference from Table 6.4</th>
<th>Level of association</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi_1 \rightarrow P$ do (Academic, $\alpha_1$)</td>
<td>Academic</td>
<td>$\alpha_1$, Create Course Notification of Intention (CNI)</td>
<td>Permission</td>
</tr>
<tr>
<td>$\phi_2 \rightarrow O$ do (head_of_school, $\alpha_2$)</td>
<td>Head of School</td>
<td>$\alpha_2$, Submit CNI</td>
<td>Obligation</td>
</tr>
<tr>
<td>$\phi_3 \rightarrow O$ do (Dean, $\alpha_3$)</td>
<td>Dean</td>
<td>$\alpha_3$, Recommend CNI</td>
<td>Obligation</td>
</tr>
<tr>
<td>$\phi_4 \rightarrow O$ do (pro_vice_chancellor_academic, $\alpha_4$)</td>
<td>pro vice chancellor academic</td>
<td>$\alpha_4$, Approve CNI</td>
<td>Obligation</td>
</tr>
<tr>
<td>$\phi_5 \rightarrow P$ do (secretary_to_VCAC, $\alpha_5$)</td>
<td>secretary to VCAC</td>
<td>$\alpha_5$, Approve CNI</td>
<td>Permission</td>
</tr>
<tr>
<td>KAT representation to present schema level constraint</td>
<td>Role of the person</td>
<td>Description</td>
<td>Action reference from Table 6.4</td>
</tr>
<tr>
<td>-----------------------------------------------------</td>
<td>--------------------</td>
<td>-------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>( \varphi_6 \rightarrow O ) do (project_manager, ( \alpha_5 ))</td>
<td>Project manager</td>
<td>Create Full Course Proposal (FCP)</td>
<td>( \alpha_5 ) - Create Full Course Proposal (FCP)</td>
</tr>
<tr>
<td>( \varphi_7 \rightarrow P ) do (academic, ( \alpha_6 ))</td>
<td>academic</td>
<td>( \alpha_6 ) - Create unit information</td>
<td>Permission</td>
</tr>
<tr>
<td>( \varphi_8 \rightarrow O ) do (head_of_school, ( \alpha_7 ))</td>
<td>Head of School</td>
<td>Submit FCP</td>
<td>Obligation</td>
</tr>
<tr>
<td>( \varphi_9 \rightarrow O ) do (Dean, ( \alpha_8 ))</td>
<td>Dean</td>
<td>( \alpha_8 ) - academic quality and recommend FCP</td>
<td>Obligation</td>
</tr>
<tr>
<td>( \varphi_{10} \rightarrow O ) do (chair_of_CEAPC, ( \alpha_9 ))</td>
<td>chair of College Education Academic and Progression (CEAPC)</td>
<td>Recommendation for FCP</td>
<td>Obligation</td>
</tr>
<tr>
<td>( \varphi_{11} \rightarrow P ) do (secretary_of_CEAPC, ( \alpha_9 ))</td>
<td>secretary of CEAPC</td>
<td>Recommendation for FCP</td>
<td>Permission</td>
</tr>
<tr>
<td>( \varphi_{12} \rightarrow O ) do (chair_of_CAAC, ( \alpha_{10} ))</td>
<td>chair of Courses Approval and Articulation Committee (CAAC)</td>
<td>Recommendation for FCP</td>
<td>Obligation</td>
</tr>
<tr>
<td>( \varphi_{13} \rightarrow P ) do (secretary_of_CAAC, ( \alpha_{10} ))</td>
<td>secretary to CAAC</td>
<td>Recommendation for FCP</td>
<td>Permission</td>
</tr>
<tr>
<td>( \varphi_{14} \rightarrow O ) do (chair_of_AS, ( \alpha_{11} ))</td>
<td>chair of Academic Senate (AS)</td>
<td>Approve FCP</td>
<td>Obligation</td>
</tr>
<tr>
<td>( \varphi_{15} \rightarrow P ) do (secretary_of_AS, ( \alpha_{11} ))</td>
<td>secretary of Academic Senate (AS)</td>
<td>Approve FCP</td>
<td>Permission</td>
</tr>
<tr>
<td>( \varphi_{16} \rightarrow O ) do (courses_officer, ( \alpha_{12} ))</td>
<td>Courses officer</td>
<td>Fill in the course code information</td>
<td>Obligation</td>
</tr>
<tr>
<td>( \varphi_{17} \rightarrow O ) do (project_manager, ( \alpha_{13} ))</td>
<td>Project Manager</td>
<td>Commence internal forum consultation</td>
<td>Obligation</td>
</tr>
<tr>
<td>( \varphi_{18} \rightarrow O ) do (associate_Dean_academic, ( \alpha_{14} ))</td>
<td>Associate Dean Academic</td>
<td>Submit internal forum consultation report</td>
<td>Obligation</td>
</tr>
<tr>
<td>( \varphi_{19} \rightarrow O ) do (project_manager, ( \alpha_{15} ))</td>
<td>Project Manager</td>
<td>Commence external advisory committee consultation</td>
<td>Obligation</td>
</tr>
<tr>
<td>( \varphi_{20} \rightarrow O ) do (associate_Dean_academic, ( \alpha_{16} ))</td>
<td>Associate Dean Academic</td>
<td>Submit external advisory committee consultation report</td>
<td>Obligation</td>
</tr>
<tr>
<td>( \varphi_{21} \rightarrow O ) do (project_manager, ( \alpha_{17} ))</td>
<td>Project Manager</td>
<td>Create and submit Resources Assessment (RA) documentation</td>
<td>Obligation</td>
</tr>
<tr>
<td>( \varphi_{22} \rightarrow O ) do (director_of_marketing, ( \alpha_{18} ))</td>
<td>Director of Marketing</td>
<td>Office of Marketing Sign-off on the RA document</td>
<td>Obligation</td>
</tr>
<tr>
<td>( \varphi_{23} \rightarrow O ) do (director_of_itd, ( \alpha_{19} ))</td>
<td>Director of ITD</td>
<td>IT directorate’s (ITD) sign-off on the RA document</td>
<td>Obligation</td>
</tr>
<tr>
<td>KAT representation to present schema level constraint</td>
<td>Role of the person</td>
<td>Action reference from Table 6.4</td>
<td>Level of association</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>--------------------</td>
<td>-------------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>$\varphi_{24} \rightarrow O$ do (director_of_edu, $\alpha_{20}$)</td>
<td>Director of EDU</td>
<td>$\alpha_{23}$ - Education Development Unit’s (EDU) sign-off on the RA document</td>
<td>Obligation</td>
</tr>
<tr>
<td>$\varphi_{25} \rightarrow O$ do (librarian, $\alpha_{21}$)</td>
<td>Librarian</td>
<td>$\alpha_{21}$ - Library sign-off on the RA document</td>
<td>Obligation</td>
</tr>
<tr>
<td>$\varphi_{26} \rightarrow O$ do (assistant_registrar_systems, $\alpha_{22}$)</td>
<td>Assistant Registrar Systems</td>
<td>$\alpha_{22}$ - Office of Academic Registrar – System’s (OAR-Sys) sign-off on the RA document</td>
<td>Obligation</td>
</tr>
<tr>
<td>$\varphi_{27} \rightarrow O$ do (assistant_registrar_operations, $\alpha_{23}$)</td>
<td>Assistant Registrar Operations</td>
<td>$\alpha_{23}$ - Office of Academic Registrar – Operation’s (OAR-opr) sign-off on the RA document</td>
<td>Obligation</td>
</tr>
<tr>
<td>$\varphi_{28} \rightarrow O$ do (college_timetabling_officer, $\alpha_{24}$)</td>
<td>College Timetabling Officer</td>
<td>$\alpha_{24}$ - Timetabling sign-off on the RA document</td>
<td>Obligation</td>
</tr>
<tr>
<td>$\varphi_{29} \rightarrow O$ do (senior_planning_architect, $\alpha_{25}$)</td>
<td>Senior Planning Architect</td>
<td>$\alpha_{25}$ - Capital Works sign-off on the RA document</td>
<td>Obligation</td>
</tr>
<tr>
<td>$\varphi_{30} \rightarrow O$ do (Dean, $\alpha_{26}$)</td>
<td>Dean</td>
<td>$\alpha_{26}$ - Approve RA document</td>
<td>Obligation</td>
</tr>
<tr>
<td>$\varphi_{31} \rightarrow O$ do (project_manager, $\alpha_{27}$)</td>
<td>Project manager</td>
<td>$\alpha_{27}$ - Create Business Case (BC) document</td>
<td>Obligation</td>
</tr>
<tr>
<td>$\varphi_{32} \rightarrow O$ do (business_development_officer, $\alpha_{28}$)</td>
<td>Business Development Officer</td>
<td>$\alpha_{28}$ - Assess and Sign-off the BC with an independent statement</td>
<td>Obligation</td>
</tr>
<tr>
<td>$\varphi_{33} \rightarrow O$ do (Dean, $\alpha_{29}$)</td>
<td>Dean</td>
<td>$\alpha_{29}$ - Recommend BC</td>
<td>Obligation</td>
</tr>
<tr>
<td>$\varphi_{34} \rightarrow O$ do (chair_of_VCAC, $\alpha_{30}$)</td>
<td>Chair of Vice Chancellor’s Advisory Committee</td>
<td>$\alpha_{30}$ - Vice Chancellor’s Advisory Committee Approval for BC</td>
<td>Obligation</td>
</tr>
<tr>
<td>$\varphi_{35} \rightarrow O$ do (DVC_development, $\alpha_{31}$)</td>
<td>Deputy Vice Chancellor Development</td>
<td>$\alpha_{31}$ - Business and Finance Committee Approval for BC</td>
<td>Obligation</td>
</tr>
<tr>
<td>$\varphi_{36} \rightarrow O$ do (Dean, $\alpha_{32}$)</td>
<td>Dean</td>
<td>$\alpha_{32}$ - College Approval for BC</td>
<td>Obligation</td>
</tr>
<tr>
<td>$\varphi_{37} \rightarrow P$ do (all, $\alpha_{33}$)</td>
<td>All</td>
<td>$\alpha_{33}$ - View course details</td>
<td>Permission</td>
</tr>
</tbody>
</table>

These guard elements $\varphi_1$ to $\varphi_{37}$ will contribute to expanding the composite guard elements $C_1$ to $C_{32}$, $C_m$ and $C_n$ (used in the KAT Expression-6.9) later in this chapter.
6.4.3 Representing Associations between Actions and Business Object using KAT

The types of associations between process actions and business object elements were identified in Chapter-5 as, *data visibility* in process actions, *data interaction* between process actions and *data-based routing* of process actions.

Definition-12 below presents the mechanism to capture the *visibility constraint* between information object and the process actions.

**Definition-12:** \( \varphi_x \rightarrow V((DATA => \{name => (reference), location => (database.table.attribute | folder.document | database.table | class.object), display => (view| edit| hidden), format => (textf | texta|selection|checkbox|radio | label | report|default)), ACTION) \)

In the above definition, the letter V identifies the visibility trait. In the argument, there are two parts. First, *data* section characterises the information element in the business process. The second section indicates the *action* to which this data is associated.

When describing the data, it uses four factors as follows.

- **Name:** This is a simple *name* or some *reference* to the data element or block, for example, *basic course information* is a table in OCAS course object, *business case information* is a report in OCAS course object and course code is an attribute in a table.

- **Location:** This helps to locate or pin point the data element or block. The location could be referred to a table at the database level, a document in a folder or an attribute in a database table. Also for referencing purposes, both relative or absolute referencing could be used as follows:
  - Relative reference such as
    
    `../ocas/documents/reports.businesscaseinformation 49.doc`
  - Absolute reference similar to
    
    `http://ocastest.uws.edu.au/web/ocas/documents/business%case%information%49.doc`

Here there is no separation made whether this data is internal or external to the workflow management system, as suggested by Sadiq et al. (2004). The main
reason behind this is that irrespective of the scope of the data, at the end they will be implemented as either a database or a document (Excel, Word, XML, Text).

- **Display**: This element identifies the basic method of displaying; viewable, editable or hidden, from the action in focus.

- **Format**: The element identifies the particular format that will be shown on the UI. The possible options are textf – text field, texta – text area, selection – section list or combo box, check boxes, radio buttons, form label or as an attached report.

When certain information elements (table, attribute, report) are to be hidden from an action interface, there are two methods to represent it. First method is to explicitly indicating that information element is required to be hidden using the display characteristic explained above. The second method is to omit the particular information element from constraint definition.

Definition-13 below presents a mechanism to capture the data interaction constraint between information object and the process actions. Certain actions in a process would depend on another to capture information that is crucial data for the process task. These data values may be requisites for formatting the UI or making routing decisions. This data interaction aspect is represented as follows;

**Definition-13**: $\phi_x \rightarrow \mathbf{I} ((\text{DATA}=> \{\text{reference1, reference2, ...}\}), \text{action x, action y})$

Above definition shows the dependency of one action (y) having to rely on another (x) for certain essential data elements to be captured, which are referenced in the data section.

Definition-14 presents a mechanism to capture the constraints definition for conditional routing.

**Definition-14**: $\phi_x \rightarrow \text{RC} (\text{variable} => \{\text{reference}\}, \text{operator} => \{\text{eq|gt|lt|el|eg}\}, \text{value} => \{\text{any|1|0|true|false|yes|no}\})$

According to Definition-14, a routing condition (RC) is built using three elements, variable, operator and value. The variable element gives the reference to a particular attribute in the information object. The operator presents a logical operative such as
equal (eq), greater than (gt), less than (lt), equal or less (el) and equal or greater (eg). The value captures the default value expected for the routing to take place. This could hold binary values such as 0 or 1, yes or no. Similarly, value element could hold any other data value such as a specific string. For example, course offering attribute value to be equal to onshore or offshore.

6.4.3.1 Application of KAT Definitions-12, 13 and 14 in OCAS

This section exemplifies the use of KAT Definitions-12, 13 and 14 to capture the data visibility in actions, data interaction between actions, and data-based routing of actions, in relation to the New Course Approval Process of OCAS.

The first action $a_1$-Create Course Notification of Intention (CNI), of the New Course Approval Process of OCAS, has visibility to three categories of information: basic course details, course responsibility and course delivery details. Each information category is captured as a separate entity in an object-relational database, with a number of attributes in each of them. Table 6.4 presents the visibility factors (name, location, display, and format) of the action $a_1$ for each information category.
Table 6.4- Description of Data visibility in action $\alpha_1$ - Create Course Notification of Intention (CNI) of the New Course Approval Process

<table>
<thead>
<tr>
<th>Information Category</th>
<th>Server location</th>
<th>Location</th>
<th>Display Methods</th>
<th>Formatting information on UI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic Course Details</strong></td>
<td><a href="http://ocastest.uws.edu.au/web/ocas/data/">http://ocastest.uws.edu.au/web/ocas/data/</a></td>
<td>cas basic_course_deatils course_name</td>
<td>editable</td>
<td>Text field</td>
</tr>
<tr>
<td></td>
<td><a href="http://ocastest.uws.edu.au/web/ocas/data/">http://ocastest.uws.edu.au/web/ocas/data/</a></td>
<td>cas basic_course_deatils course_code</td>
<td>hidden</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td><a href="http://ocastest.uws.edu.au/web/ocas/data/">http://ocastest.uws.edu.au/web/ocas/data/</a></td>
<td>cas basic_course_deatils course_duration</td>
<td>editable</td>
<td>Text field</td>
</tr>
<tr>
<td></td>
<td><a href="http://ocastest.uws.edu.au/web/ocas/data/">http://ocastest.uws.edu.au/web/ocas/data/</a></td>
<td>cas basic_course_deatils course_type (undergraduate, postgraduate)</td>
<td>editable</td>
<td>selection $\rightarrow$ transfer from another table</td>
</tr>
<tr>
<td></td>
<td><a href="http://ocastest.uws.edu.au/web/ocas/data/">http://ocastest.uws.edu.au/web/ocas/data/</a></td>
<td>cas basic_course_deatils course_offering (onshore or offshore)</td>
<td>editable</td>
<td>selection $\rightarrow$ transfer from another table</td>
</tr>
<tr>
<td></td>
<td><a href="http://ocastest.uws.edu.au/web/ocas/data/">http://ocastest.uws.edu.au/web/ocas/data/</a></td>
<td>cas basic_course_deatils course_description</td>
<td>editable</td>
<td>Text field</td>
</tr>
<tr>
<td><strong>Course Responsibility</strong></td>
<td><a href="http://ocastest.uws.edu.au/web/ocas/data/">http://ocastest.uws.edu.au/web/ocas/data/</a></td>
<td>cas course_responsibility college</td>
<td>editable</td>
<td>selection $\rightarrow$ transfer from another table</td>
</tr>
<tr>
<td></td>
<td><a href="http://ocastest.uws.edu.au/web/ocas/data/">http://ocastest.uws.edu.au/web/ocas/data/</a></td>
<td>cas course_responsibility school</td>
<td>editable</td>
<td>selection $\rightarrow$ transfer from another table</td>
</tr>
<tr>
<td></td>
<td><a href="http://ocastest.uws.edu.au/web/ocas/data/">http://ocastest.uws.edu.au/web/ocas/data/</a></td>
<td>cas course_responsibility key_contact_person</td>
<td>editable</td>
<td>selection $\rightarrow$ transfer from another table</td>
</tr>
<tr>
<td></td>
<td><a href="http://ocastest.uws.edu.au/web/ocas/data/">http://ocastest.uws.edu.au/web/ocas/data/</a></td>
<td>cas course_responsibility course_coordinator</td>
<td>editable</td>
<td>selection $\rightarrow$ transfer from another table</td>
</tr>
<tr>
<td><strong>Course Delivery Details</strong></td>
<td><a href="http://ocastest.uws.edu.au/web/ocas/data/">http://ocastest.uws.edu.au/web/ocas/data/</a></td>
<td>cas course_delivery_details delivery_mode (external, part time, full time, online)</td>
<td>editable</td>
<td>selection $\rightarrow$ transfer from another table</td>
</tr>
<tr>
<td></td>
<td><a href="http://ocastest.uws.edu.au/web/ocas/data/">http://ocastest.uws.edu.au/web/ocas/data/</a></td>
<td>cas course_delivery_details Location (campuses)</td>
<td>editable</td>
<td>selection $\rightarrow$ transfer from another table</td>
</tr>
<tr>
<td></td>
<td><a href="http://ocastest.uws.edu.au/web/ocas/data/">http://ocastest.uws.edu.au/web/ocas/data/</a></td>
<td>cas course_delivery_details delivery_type (lecture, practical, tutorial)</td>
<td>editable</td>
<td>selection $\rightarrow$ transfer from another table</td>
</tr>
<tr>
<td></td>
<td><a href="http://ocastest.uws.edu.au/web/ocas/data/">http://ocastest.uws.edu.au/web/ocas/data/</a></td>
<td>cas course_delivery_details number_ftof_hours</td>
<td>editable</td>
<td>selection $\rightarrow$ transfer from another table</td>
</tr>
</tbody>
</table>
When closely analysing the data in Table 6.4, it is clear that all attributes in these three information categories (basic course details, course responsibility and course delivery details) are editable except for a special attribute named course_code under basic course details. Course code details are entered in a different action by a different participant; therefore it is not visible in this first action. Further, according to Table 6.4, the attributes in information artefacts course responsibility and course delivery details have the same type of display and formatting methods. Therefore, instead of representing each attribute, these can be presented at entity level using the Definition-12. However, the attributes in basic course details artefacts have different types of display and formatting methods; thus required to be represented at attribute level.

Based on the information in Table 6.4 and Definition-12, the visibility factors of action $\alpha_1$-Create Course Notification of Intention (CNI) is written in the guard element $\varphi_{38}$ as follows:

$$\varphi_{38} \rightarrow V(X\rightarrow \{ \text{name} => (\text{course_name}), \text{location} => (\text{http://ocastest.uws.edu.au/web/ocas/data/ocas.basic_course_details.course_name}), \text{display} => (\text{edit}), \text{format} => (\text{textf}), \{ \text{name} => (\text{course_code}), \text{location} => (\text{http://ocastest.uws.edu.au/web/ocas/data/ocas.basic_course_details.course_code}), \text{display} => (\text{hidden}), \{ \text{name} => (\text{course_duration}), \text{location} => (\text{http://ocastest.uws.edu.au/web/ocas/data/ocas.basic_course_details.course_duration}), \text{display} => (\text{edit}), \text{format} => (\text{textf}), \{ \text{name} => (\text{course_type}), \text{location} => (\text{http://ocastest.uws.edu.au/web/ocas/data/ocas.basic_course_details.course_type}), \text{display} => (\text{edit}), \text{format} => (\text{select}), \} \}, \{ \text{name} => (\text{course_offering}), \text{location} => (\text{http://ocastest.uws.edu.au/web/ocas/data/ocas.basic_course_details.course_offering}), \text{display} => (\text{edit}), \text{format} => (\text{select}), \} \}, \{ \text{name} => (\text{course_description}), \text{location} => (\text{http://ocastest.uws.edu.au/web/ocas/data/ocas.basic_course_details.course_description}), \text{display} => (\text{edit}), \text{format} => (\text{textf}), \} \}, \{ \text{name} => (\text{course_responsibility}), \text{location} => (\text{http://ocastest.uws.edu.au/web/ocas/data/ocas.course_responsibility}), \text{display} => (\text{edit}), \text{format} => (\text{select}), \} \}, \{ \text{name} => (\text{course_delivery_details}), \text{location} => (\text{http://ocastest.uws.edu.au/web/ocas/data/ocas.course_delivery_details}), \text{display} => (\text{edit}), \text{format} => (\text{select}) \} \})$$

Similar to action $\alpha_1$ and its associated visibility guard element $\varphi_{38}$, other actions $\alpha_2$ to $\alpha_{33}$ (listed in Table 6.2) have the visibility aspect defined into a set of guard elements. Here those guard elements, associated with actions $\alpha_2$ to $\alpha_{33}$, are not explicitly defined as $\varphi_{38}$ above, due to its repetitive nature. However, it is assumed that the notations $\varphi_{39}$ to $\varphi_{70}$ represent the visibility constraints associated with actions $\alpha_2$ to $\alpha_{33}$ respectively.

The next aspect of associations between actions and business object is the data interaction between actions. This refers to the concept of one action depending
on another to capture certain data elements. Before defining this aspect, first it is required to identify the action interfaces, in which the business object data is first entered into the systems. Table 6.5 below extracts the business object elements and actions in which data are entered.

**Table 6.5- Guard Conditions of the Action in which Course Information is initially entered in the New Course Approval Process**

<table>
<thead>
<tr>
<th>Course Business Object Element</th>
<th>Action in which this data is initially entered</th>
<th>Associated guard constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic course details</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Course code</td>
<td>α₁₂ - Fill in the course code information</td>
<td>φ₄₉</td>
</tr>
<tr>
<td>Course name</td>
<td>α₁ - Create Course Notification of Intention</td>
<td>φ₃₈</td>
</tr>
<tr>
<td>Course Duration</td>
<td>α₁ - Create Course Notification of Intention</td>
<td>φ₃₈</td>
</tr>
<tr>
<td>Course Type</td>
<td>α₁ - Create Course Notification of Intention</td>
<td>φ₃₈</td>
</tr>
<tr>
<td>Course Offering</td>
<td>α₁ - Create Course Notification of Intention</td>
<td>φ₃₈</td>
</tr>
<tr>
<td>Course description</td>
<td>α₁ - Create Course Notification of Intention</td>
<td>φ₃₈</td>
</tr>
<tr>
<td><strong>Course responsibility</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name of the College</td>
<td>α₁ - Create Course Notification of Intention</td>
<td>φ₃₈</td>
</tr>
<tr>
<td>Name of the School</td>
<td>α₁ - Create Course Notification of Intention</td>
<td>φ₃₈</td>
</tr>
<tr>
<td>Key contact person</td>
<td>α₁ - Create Course Notification of Intention</td>
<td>φ₃₈</td>
</tr>
<tr>
<td>Course Coordinator</td>
<td>α₁ - Create Course Notification of Intention</td>
<td>φ₃₈</td>
</tr>
<tr>
<td><strong>Course delivery details</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mode of delivery</td>
<td>α₁ - Create Course Notification of Intention</td>
<td>φ₃₈</td>
</tr>
<tr>
<td>Location</td>
<td>α₁ - Create Course Notification of Intention</td>
<td>φ₃₈</td>
</tr>
<tr>
<td>Type of delivery</td>
<td>α₁ - Create Course Notification of Intention</td>
<td>φ₃₈</td>
</tr>
<tr>
<td>Number of Face to face hours</td>
<td>α₁ - Create Course Notification of Intention</td>
<td>φ₃₈</td>
</tr>
<tr>
<td><strong>Course Structure</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Key programs</td>
<td>α₅ - Create Full Course Proposal (FCP)</td>
<td>φ₄₂</td>
</tr>
<tr>
<td>Units associated with key programs</td>
<td>α₅ - Create Full Course Proposal (FCP)</td>
<td>φ₄₂</td>
</tr>
<tr>
<td>Unit details</td>
<td>α₆ - Create unit information</td>
<td>φ₄₃</td>
</tr>
<tr>
<td>Recommended progression</td>
<td>α₅ - Create Full Course Proposal (FCP)</td>
<td>φ₄₂</td>
</tr>
<tr>
<td><strong>Graduate Attributes</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>University graduate attributes</td>
<td>α₅ - Create Full Course Proposal (FCP)</td>
<td>φ₄₂</td>
</tr>
<tr>
<td>Course graduate attributes</td>
<td>α₅ - Create Full Course Proposal (FCP)</td>
<td>φ₄₂</td>
</tr>
<tr>
<td><strong>Course Resource Details</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Staffing requirements</td>
<td>α₁₇ - Create and submit Resources Assessment (RA) documentation</td>
<td>φ₅₄</td>
</tr>
<tr>
<td>Associations with other organisations</td>
<td>α₁₇ - Create and submit Resources Assessment (RA) documentation</td>
<td>φ₅₄</td>
</tr>
<tr>
<td>Library facility requirements</td>
<td>α₁₇ - Create and submit Resources Assessment (RA) documentation</td>
<td>φ₅₄</td>
</tr>
<tr>
<td>Rooms and lab requirements</td>
<td>α₁₇ - Create and submit Resources Assessment (RA) documentation</td>
<td>φ₅₄</td>
</tr>
<tr>
<td>Special training and skills requirements</td>
<td>α₁₇ - Create and submit Resources Assessment (RA) documentation</td>
<td>φ₅₄</td>
</tr>
<tr>
<td><strong>Course business case details</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Competitor information</td>
<td>α₂₇ - Create Business Case (BC) document</td>
<td>φ₆₄</td>
</tr>
<tr>
<td>Potential market information</td>
<td>α₂₇ - Create Business Case (BC) document</td>
<td>φ₆₄</td>
</tr>
<tr>
<td>Course risk factors</td>
<td>α₂₇ - Create Business Case (BC) document</td>
<td>φ₆₄</td>
</tr>
<tr>
<td>Estimated costs</td>
<td>α₂₇ - Create Business Case (BC) document</td>
<td>φ₆₄</td>
</tr>
<tr>
<td>Course Business Object Element</td>
<td>Main Attributes</td>
<td>Action in which this data is initially entered</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>Information component</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Estimated revenue</td>
<td>α_{27} - Create Business Case (BC) document</td>
<td>φ_{64}</td>
</tr>
<tr>
<td>Course Marketing information</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Potential Career paths</td>
<td>α_{5} - Create Full Course Proposal (FCP)</td>
<td>φ_{42}</td>
</tr>
<tr>
<td>Enrolment requirements</td>
<td>α_{5} - Create Full Course Proposal (FCP)</td>
<td>φ_{42}</td>
</tr>
<tr>
<td>Enrolment restrictions</td>
<td>α_{5} - Create Full Course Proposal (FCP)</td>
<td>φ_{42}</td>
</tr>
<tr>
<td>Consultation Feedback</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internal Forum Consultation feedback</td>
<td>α_{14} - Submit internal forum consultation report</td>
<td>φ_{51}</td>
</tr>
<tr>
<td>External Advisory Committee Consultation Feedback</td>
<td>α_{15} - Commence external advisory committee consultation</td>
<td>φ_{52}</td>
</tr>
</tbody>
</table>

In OCAS related *New Course Approval Process*, there are a number of data interaction dependencies among actions. Table 6.6 shows these dependencies. The column constraint notation presents data interaction dependency among actions according to Definition-13. The rest of the columns show the elements of this notational representation.

In Table 6.6 the constraints φ_{77}, φ_{89}, φ_{90} and φ_{91} are a special set of constraints, which are later used for action routing.
# Table 6.6- Data Interaction Dependency in the New Course Approval Process

<table>
<thead>
<tr>
<th>Constraint notation</th>
<th>Data Elements</th>
<th>Action in which data is captured (action $x$ in definitions 13)</th>
<th>Action in which this data is used (action $y$ in Definition-13)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varphi_7 \rightarrow I((\text{data}={\text{course}_\text{name}}, \alpha_1, \alpha_5))$</td>
<td>course_name - Course name</td>
<td>$\alpha_5$ - Create Course Notification of Intention</td>
<td>$\alpha_5$ - Create Full Course Proposal (FCP)</td>
</tr>
<tr>
<td>$\varphi_7 \rightarrow I((\text{data}={\text{course}<em>\text{name}}, \alpha_1, \alpha</em>{12}))$</td>
<td>course_name - Course name</td>
<td>$\alpha_5$ - Create Course Notification of Intention</td>
<td>$\alpha_{12}$ - Fill in the course code information</td>
</tr>
<tr>
<td>$\varphi_7 \rightarrow I((\text{data}={\text{course}<em>\text{name}}, \alpha_1, \alpha</em>{13}))$</td>
<td>course_name - Course name</td>
<td>$\alpha_5$ - Create Course Notification of Intention</td>
<td>$\alpha_{13}$ - Commence internal forum consultation</td>
</tr>
<tr>
<td>$\varphi_7 \rightarrow I((\text{data}={\text{course}<em>\text{name}}, \alpha_1, \alpha</em>{15}))$</td>
<td>course_name - Course name</td>
<td>$\alpha_5$ - Create Course Notification of Intention</td>
<td>$\alpha_{15}$ - Commence external advisory committee consultation</td>
</tr>
<tr>
<td>$\varphi_7 \rightarrow I((\text{data}={\text{course}<em>\text{name}}, \alpha_1, \alpha</em>{17}))$</td>
<td>course_name - Course name</td>
<td>$\alpha_5$ - Create Course Notification of Intention</td>
<td>$\alpha_{27}$ - Create and submit Resources Assessment (RA) documentation</td>
</tr>
<tr>
<td>$\varphi_7 \rightarrow I((\text{data}={\text{course}<em>\text{name}}, \alpha_1, \alpha</em>{27}))$</td>
<td>course_name - Course name</td>
<td>$\alpha_5$ - Create Course Notification of Intention</td>
<td>$\alpha_{27}$ - Create Business Case (BC) document</td>
</tr>
<tr>
<td>$\varphi_7 \rightarrow I((\text{data}={\text{units}<em>\text{key}</em>\text{prog}}, \alpha_5, \alpha_6))$</td>
<td>units_key_prog - Units associated with key programs</td>
<td>$\alpha_5$ - Create Full Course Proposal (FCP)</td>
<td>$\alpha_6$ - Create unit information</td>
</tr>
<tr>
<td>$\varphi_7 \rightarrow I((\text{data}={\text{uni}<em>\text{grad}</em>\text{attrb}, \text{course}<em>\text{grad}</em>\text{attrb}, \text{key}<em>\text{progs}, \text{units}</em>\text{key}_\text{prog}}, \alpha_5, \alpha_6))$</td>
<td>uni_grad_attrb - University graduate attributes, course_grad_attrb - Course graduate attributes, key_progs - Key programs, units_key_prog - Units associated with key programs</td>
<td>$\alpha_5$ - Create Full Course Proposal (FCP)</td>
<td>$\alpha_8$ - Assess academic quality and recommend FCP</td>
</tr>
<tr>
<td>$\varphi_7 \rightarrow I((\text{data}={\text{career}<em>\text{path}, \text{enroll}</em>\text{requ}, \text{enroll}<em>\text{restric}}, \alpha_5, \alpha</em>{18}))$</td>
<td>career_path - Potential career paths, enroll_requ - Enrolment requirements, enroll_restric - Enrolment restrictions</td>
<td>$\alpha_5$ - Create Full Course Proposal (FCP)</td>
<td>$\alpha_{28}$ - Office of Marketing Sign-off on the RA document</td>
</tr>
<tr>
<td>$\varphi_7 \rightarrow I((\text{data}={\text{room}<em>\text{lab}}, \alpha</em>{17}, \alpha_{19}))$</td>
<td>room_lab - Rooms and lab requirements</td>
<td>$\alpha_{17}$ - Create and submit Resources Assessment (RA) documentation</td>
<td>$\alpha_{19}$ - IT directorate’s (ITD) sign-off on the RA document</td>
</tr>
<tr>
<td>$\varphi_{94} \rightarrow I((\text{data}={\text{train}<em>\text{skill}</em>\text{req}})$</td>
<td>train_skill_req - Special training and</td>
<td>$\alpha_{17}$ - Create and submit Resources Assessment (RA) documentation</td>
<td>$\alpha_{20}$ - Education Development Unit’s (EDU)</td>
</tr>
<tr>
<td>Constraint notation</td>
<td>Data Elements</td>
<td>Description</td>
<td>Action in which data is captured (action x in definitions 13)</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------</td>
<td>-------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>{train_skill_req}, α_{17}, α_{20}</td>
<td>skills requirements</td>
<td>Assessment (RA) documentation</td>
<td>sign-off on the RA document</td>
</tr>
<tr>
<td>(\varphi_{82}) (\rightarrow) (I) ((\text{data=&gt; {lib_facility }}, α_{17}, α_{21})</td>
<td>lib_facility - Library facility requirements</td>
<td>α_{17} - Create and submit Resources Assessment (RA) documentation</td>
<td>α_{21} - Library sign-off on the RA document</td>
</tr>
<tr>
<td>(\varphi_{83}) (\rightarrow) (I) ((\text{data=&gt; {enroll_requ, enroll_restric }}, α_{17}, α_{22})</td>
<td>enroll_requ - Enrolment requirements, enroll_restric - Enrolment restrictions</td>
<td>α_{17} - Create and submit Resources Assessment (RA) documentation</td>
<td>α_{22} - Office of Academic Registrar – System’s (OAR-Sys) sign-off on the RA document</td>
</tr>
<tr>
<td>(\varphi_{84}) (\rightarrow) (I) ((\text{data=&gt; {course_duration,course_type, course_offering }}, α_{1}, α_{22})</td>
<td>course_duration - Course Duration, course_type - Course Type, course_offering - Course Offering</td>
<td>α_{1} - Create Course Notification of Intention</td>
<td>α_{22} - Office of Academic Registrar – System’s (OAR-Sys) sign-off on the RA document</td>
</tr>
<tr>
<td>(\varphi_{85}) (\rightarrow) (I) ((\text{data=&gt; {enroll_requ, enroll_restric }}, α_{17}, α_{23})</td>
<td>enroll_requ - Enrolment requirements, enroll_restric - Enrolment restrictions</td>
<td>α_{17} - Create and submit Resources Assessment (RA) documentation</td>
<td>α_{23} - Office of Academic Registrar – Operation’s (OAR-opr) sign-off on the RA document</td>
</tr>
<tr>
<td>(\varphi_{86}) (\rightarrow) (I) ((\text{data=&gt; {delivery_mode, location, delivery_type, num_ftof_hours }}, α_{1}, α_{24})</td>
<td>delivery_mode - Mode of delivery, location - Location, delivery_type - Type of delivery, num_ftof_hours - Number of Face to face hours</td>
<td>α_{1} - Create Course Notification of Intention</td>
<td>α_{24} - Timetabling sign-off on the RA document</td>
</tr>
<tr>
<td>(\varphi_{87}) (\rightarrow) (I) ((\text{data=&gt; {room_lab}}, α_{17}, α_{24})</td>
<td>room_lab - Rooms and lab requirements</td>
<td>α_{17} - Create and submit Resources Assessment (RA) documentation</td>
<td>α_{24} - Timetabling sign-off on the RA document</td>
</tr>
<tr>
<td>(\varphi_{88}) (\rightarrow) (I) ((\text{data=&gt; {staff_req, room_lab }}, α_{17}, α_{25})</td>
<td>staff_req - Staffing requirements, room_lab - Rooms and lab requirements</td>
<td>α_{17} - Create and submit Resources Assessment (RA) documentation</td>
<td>α_{25} - Capital Works sign-off on the RA document</td>
</tr>
<tr>
<td>(\varphi_{89}) (\rightarrow) (I) ((\text{data=&gt; {course_offering }}, α_{1}, α_{30})</td>
<td>course_offering - Course offering</td>
<td>α_{1} - Create Course Notification of Intention</td>
<td>α_{30} - Vice Chancellor’s Advisory Committee Approval for BC</td>
</tr>
<tr>
<td>(\varphi_{90}) (\rightarrow) (I) ((\text{data=&gt; {course_offering }}, α_{1}, α_{31})</td>
<td>course_offering - Course offering</td>
<td>α_{1} - Create Course Notification of Intention</td>
<td>α_{31} - Business and Finance Committee Approval for BC</td>
</tr>
<tr>
<td>(\varphi_{91}) (\rightarrow) (I) ((\text{data=&gt; {course_offering }}, α_{1}, α_{32})</td>
<td>course_offering - Course offering</td>
<td>α_{1} - Create Course Notification of Intention</td>
<td>α_{32} - College Approval for BC</td>
</tr>
</tbody>
</table>
In OCAS there are a few occasions in which certain attribute values are required to be checked for process routing. Table 6.7 illustrates these constraint definitions. The first column gives the constraint notation according to Definition-14 and rest of the columns describes the elements in the constraint notation.

<table>
<thead>
<tr>
<th>Constraint notation</th>
<th>Description</th>
<th>Data Element or variable in the course object</th>
<th>Logical operator</th>
<th>Value that requires to be held for routing</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \phi_{92} \rightarrow \text{RC}(\text{variable} \Rightarrow { \text{units}<em>\text{key}</em>\text{prog} }, \text{operator} \Rightarrow { \text{eq} }, \text{value} \Rightarrow { \text{no}<em>\text{of}</em>\text{units} }) )</td>
<td>units_key_prog - Units associated with key programs</td>
<td>Equal</td>
<td>Number of units contained within the course (for example, “24”)</td>
<td></td>
</tr>
<tr>
<td>( \phi_{93} \rightarrow \text{RC}(\text{variable} \Rightarrow { \text{course}_\text{offering} }, \text{operator} \Rightarrow { \text{eq} }, \text{value} \Rightarrow { \text{Exception} }) )</td>
<td>course_offering - Course offering</td>
<td>Equal</td>
<td>“Exception”</td>
<td></td>
</tr>
<tr>
<td>( \phi_{94} \rightarrow \text{RC}(\text{variable} \Rightarrow { \text{course}_\text{offering} }, \text{operator} \Rightarrow { \text{eq} }, \text{value} \Rightarrow { \text{offshore} }) )</td>
<td>course_offering - Course offering</td>
<td>Equal</td>
<td>“offshore”</td>
<td></td>
</tr>
<tr>
<td>( \phi_{95} \rightarrow \text{RC}(\text{variable} \Rightarrow { \text{course}_\text{offering} }, \text{operator} \Rightarrow { \text{eq} }, \text{value} \Rightarrow { \text{onshore} }) )</td>
<td>course_offering - Course offering</td>
<td>Equal</td>
<td>“onshore”</td>
<td></td>
</tr>
</tbody>
</table>

In this section, three newly introduced KAT Definitions-12, 13 and 14 were applied to represent the associations between actions and data in the New Course Approval Process of OCAS. The outcome is, an addition of phi (guard) elements (\( \phi_{38} \) to \( \phi_{95} \)) to the set continuing from previous sections. These phis will later contribute to expand the composite conditions \( C_1 \) to \( C_{32}, C_m \) and \( C_n \) (used in the KAT Expression-6.9).

### 6.4.4 Representing Associations among Business Object Elements using KAT

*Integrity* and *computational* relationships are the two types of associations that exist within business object elements.

Definition-15 introduces a mechanism to present *integrity* relationships among information attributes.

**Definition-15:** \( \phi_x \rightarrow \text{IC(} \text{subject component} \Rightarrow \{ \text{reference} \}, \text{related components} \Rightarrow \{ \text{reference1, reference2 , ...} \}, \text{connector} \Rightarrow \{ \text{reference, reference, .} \}, \text{type of link} \Rightarrow \{ \text{foreign key | hyperlink | document link } \} \) \)

According to Definition-15, integrity constraints (IC) are presented using four elements *subject component*, *related components*, *connector*, and *type of link*.
The subject component element is the subject of this definition; as the association is defined in relation to this element. The subject component may refer to an attribute, a document or a table as appropriate with the process implementation. The related components refer to other informational components that are related to the subject component. Similar to subject component, the related components could refer to attributes or tables. The connector refers to the common data element used between subject component and related components to identify the relationship. For example, a common attribute used in primary and foreign key relationship in a relational database. It is possible for more than one data item to be used to define the relationship. The type of link defines the mechanism used to establish this relationship. For example, it could be hyperlinks between two documents or a foreign key relationship between two database tables.

Definition-16 represents the integrity constraints among information attributes.

**Definition-16:** \( \varphi_x \rightarrow \text{CC (subject component} \Rightarrow \{\text{reference}\}, \text{related components} \Rightarrow \{\text{reference1, reference2, ...}\}, \text{type of computation} \Rightarrow \{\text{summation | average | other formula}\})\)

The computational constraint (CC) is defined using three elements, subject component, related components and type of computation in Definition-16. The use of subject component and related components are similar to the above Definition-15. The type of computation refers to the mechanism in which the computation is carried out. For example, the related components could be totalled to create the subject component. In that case, the type of computation is a summation. If the computation is not simple as a summation or average, it is possible to include the formula that is used to perform the computation.

### 6.4.4.1 Application of KAT Definitions-15 and 16 in OCAS

In OCAS related New Course Approval Process, there are certain integrity and computational constraints associated with its information object- course. The integrity constraints are defined in Table 6.8 and computational constraints are illustrated in Table 6.9. In both tables, initial column gives the constraint notation.
according to the Definition-15 and 16 above. The rest of the columns explain the elements that constitute the constraint notations.

These tables identify four more phi (guard) elements ($\phi_{96}$ to $\phi_{99}$) that get added to the running list of constraints related to the New Course Approval Process in OCAS.
### Table 6.8 - The Integrity Constraints Associated with the Information Object of the New Course Approval Process

<table>
<thead>
<tr>
<th>Constraint notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi_{06}$ $\rightarrow$ IC (subject component $\Rightarrow$ {basic_course_details}, related components $\Rightarrow$ {course_delivery_details, course_structure, graduate_attributes, course_resource_details, course_marketing_information}, connector $\Rightarrow$ {course_code, course_name}, type of link $\Rightarrow$ {foreign key})</td>
<td>Basic Course Details</td>
</tr>
<tr>
<td>$\phi_{07}$ $\rightarrow$ IC (subject component $\Rightarrow$ {basic_course_details}, related components $\Rightarrow$ {course_business_case_details, consultation_feedback}, connector $\Rightarrow$ {course_code, course_name, document_name}, type of link $\Rightarrow$ {document link})</td>
<td>Basic Course Details</td>
</tr>
</tbody>
</table>
Table 6.9- Computational Constraints among information attributes in course object of the *New Course Approval Process*

<table>
<thead>
<tr>
<th>Constraint notation</th>
<th>Description</th>
<th>subject component</th>
<th>related components</th>
<th>Type of computation</th>
</tr>
</thead>
</table>
| \(\varphi_{98}\) \(\xrightarrow{\text{IC}}\) \{subject component \(\Rightarrow\) \{course_business_case_details.estimated_revenue\},
related components \(\Rightarrow\) \{course_business_case_details.potential_market.no_full_fee_paying_students, course_business_case_details.potential_market.no_international_students, course_business_case_details.potential_market.no_gov_funded_students, course_business_case_details.course_fee.full_fee_local, course_business_case_details.course_fee.full_fee_international, course_business_case_details.course_fee.gov_funded_stu_fee\},
type of computation \(\Rightarrow\) \{no_full_fee_paying_students \times full_fee_local + no_international_students \times full_fee_international + no_gov_funded_students \times gov_funded_stu_fee\}\} | Estimated revenue | Student numbers
Local full fee paying
International
Local government funded
Proposed Fees
Local full fee
International fee
Government funded student fee | Number of students in each category (local full fee, international and local government funded) is multiplied by the proposed fee in that category. Then these three figures are totalled to calculate the estimated revenue of the course. |
| \(\varphi_{99}\) \(\xrightarrow{\text{IC}}\) \{subject component \(\Rightarrow\) \{course_business_case_details.estimated_costs\},
related components \(\Rightarrow\) \{course_resource_details.staffing_costs, course_resource_details.facilities_costs, course_resource_details.library_costs, course_resource_details.training_costs, course_resource_details.external_costs, course_resource_details.overheads, course_business_case_details.estimated_revenue\},
type of computation \(\Rightarrow\) \{staffing_costs + facilities_costs + library_costs + training_costs + external_costs + overheads + estimated_income \times 5\%}\} | Estimated costs | Direct costs
staffing_costs
facilities_costs
library_costs
training_costs
external_costs
overheads
taxes and levies
estimated_income \times 5\% | All the direct cost figures are added. The taxes and levies are calculated as 5% of the estimated revenue and added to the direct costs |
6.4.5 Representing Constraints on the Associations among Actions using KAT

Previously in Chapter-5 the constraints on the association within process actions were identified to be *work item failures, time constraints, participant or data unavailability, external triggers* and *other constraints triggers*. Some researchers (Russell et al., 2006b) name these constraints as exceptions.

There are two types of exceptions: known and unknown. Known exceptions can be identified in priori design and can be planned for. For example, time constraints such as deadlines and certain external triggers such as other process actions. However, there are other exceptions, which are unanticipated and could not be planned for at design time. These kinds of exceptions are *work item failures* and *participant or data unavailability*. Therefore, these types of exceptions cannot be modelled in a formalism, but require a different approach to handle them. Exploration for a method to handle unknown exceptions is considered to be beyond the scope of this research. Therefore, KAT based guard elements are only defined for time or temporal constraints and external triggers that arise from external events.

Definition-17 below presents formalism to capture the temporal (or time) constraints (TC) that may affect the association between process participants and actions.

**Definition-17:** \( \varphi_x \rightarrow \text{TC} \)  
(time reference \( \rightarrow \) \{absolute | relative\}, start time \( \rightarrow \) \{seconds: minutes: hour: day: month: year\} | relative time\}, end time \( \rightarrow \) \{ seconds: minutes: hour: day: month: year\} | time period\})

According to Definition-17, there are three elements; *time reference, start time* and *end time*, used to define a temporal constraint.

There are two methods to define a deadline, use of *absolute* or *relative* references (Marjanovic, 2000). For example, a deadline could be defined either with *absolute time*, such as 11.03am: 18:05:2004, or as *relative time* period such as 34 hours. For the attribute, *time reference* in the above definitions a selection needs to be made between these two. The *start time* element allows a starting time to be specified precisely to the second using the format seconds: minutes: hour: day:
month: year. The other mechanism is to define it using relative time to the local time of the computer. For example, an allocation of a particular action to a participant could be deferred for two days, relative to the current system time.

The end time refers to the time when the deadline ends. This could be presented similar to the start time using absolute or relative time measure. If the end time is defined using relative time, the deadline is the end time. If the deadline was given in absolute time, it is calculated based on the start time and the absolute time. If the deadline is reached, this constraint or the guard element is invalid. Thus, this would create an exception that need to be handled appropriately, such as escalation of tasks (Russell et al., 2006b).

Definition-18 below presents a formalism to capture the external constraints (XC) that may affect the association between process participants and actions.

**Definition-18:** \( \varphi_x \rightarrow XC (EXTERNAL ENTITY \Rightarrow \{process \mid data \ entity \mid external \ apparatus\}, TRIGGER = \{event\Rightarrow \{affecting \ factor\}, operator \Rightarrow \{eq| gt| lt| el| eg | in | not in | yes | no | is | is not\}, value\Rightarrow \{value \ of \ the \ affecting \ factor\}) \)

In Definition-18, first a reference is made to the external entity. This external entity could be another process, a particular data entity, or an external apparatus such as thermostat that checks the external temperature. Next, the trigger that constrains the process is defined using three attributes; event, operator and value. For example, if the external entity is a thermostat that continuously feeds temperature to a system, the triggering event refers to the temperature factor. With the operator set to eq (equal) and value to 30\( ^\circ \)f, an external constraint can be created to guard a particular action.

### 6.4.5.1 Application of KAT Definitions-17 and 18 in OCAS

In OCAS related processes, there are no temporal constraints imposed, as these processes are not time critical. Therefore, there are no constraints identified in the OCAS New Course Approval Process under Definition-17.

In the New Course Approval Process, there are certain process actions that get triggered by the state of other sub-process. Figure 6.6 depicts four sub-process triggers present in the New Course Approval Process of OCAS.
(iii) CEAPC approval for course proposal is constrained until the completion of the course code entry, internal forum, external advisory committee, and recourse assessment processes.

(iv) Dean's approval for the course proposal is constrained until the completion of the business case process.

(i) Final approval of Resource Assessment is constrained until all sub-divisional sign-offs are completed.

(ii) Start of Business Case process is constrained until the completion of the RA Process.

Figure 6.6- Sub Process Triggers that affect the Association between Actions in the New Course Approval Process of OCAS.
Using Definition-18, the four sub-process triggers in Figure 6.6 are given in constraint notations $\Phi_{100}$ to $\Phi_{112}$ as follows. These constraint notation numbering continue from the previous section.

- $\Phi_{100} \rightarrow XC (\text{External Entity} \Rightarrow \{\text{library resource sign off process}\}, \text{Trigger} = (\text{event}=>\{\text{state}\}, \text{operator} => \{\text{is}\}, \text{value} => \{\text{end}\}))$
- $\Phi_{101} \rightarrow XC (\text{External Entity} \Rightarrow \{\text{ITD resource sign off process}\}, \text{Trigger} = (\text{event}=>\{\text{state}\}, \text{operator} => \{\text{is}\}, \text{value} => \{\text{end}\}))$
- $\Phi_{102} \rightarrow XC (\text{External Entity} \Rightarrow \{\text{EDU resource sign off process}\}, \text{Trigger} = (\text{event}=>\{\text{state}\}, \text{operator} => \{\text{is}\}, \text{value} => \{\text{end}\}))$
- $\Phi_{103} \rightarrow XC (\text{External Entity} \Rightarrow \{\text{Officer of Marketing resource sign off process}\}, \text{Trigger} = (\text{event}=>\{\text{state}\}, \text{operator} => \{\text{is}\}, \text{value} => \{\text{end}\}))$
- $\Phi_{104} \rightarrow XC (\text{External Entity} \Rightarrow \{\text{Timetabling resource sign off process}\}, \text{Trigger} = (\text{event}=>\{\text{state}\}, \text{operator} => \{\text{is}\}, \text{value} => \{\text{end}\}))$
- $\Phi_{105} \rightarrow XC (\text{External Entity} \Rightarrow \{\text{OAR-Systems resource sign off process}\}, \text{Trigger} = (\text{event}=>\{\text{state}\}, \text{operator} => \{\text{is}\}, \text{value} => \{\text{end}\}))$
- $\Phi_{106} \rightarrow XC (\text{External Entity} \Rightarrow \{\text{OAR-operations resource sign off process}\}, \text{Trigger} = (\text{event}=>\{\text{state}\}, \text{operator} => \{\text{is}\}, \text{value} => \{\text{end}\}))$
- $\Phi_{107} \rightarrow XC (\text{External Entity} \Rightarrow \{\text{Capital Works resource sign off process}\}, \text{Trigger} = (\text{event}=>\{\text{state}\}, \text{operator} => \{\text{is}\}, \text{value} => \{\text{end}\}))$
- $\Phi_{108} \rightarrow XC (\text{External Entity} \Rightarrow \{\text{Resource Assessment Approval process}\}, \text{Trigger} = (\text{event}=>\{\text{state}\}, \text{operator} => \{\text{is}\}, \text{value} => \{\text{end}\}))$
- $\Phi_{109} \rightarrow XC (\text{External Entity} \Rightarrow \{\text{Course Code Entry Process}\}, \text{Trigger} = (\text{event}=>\{\text{state}\}, \text{operator} => \{\text{is}\}, \text{value} => \{\text{end}\}))$
- $\Phi_{110} \rightarrow XC (\text{External Entity} \Rightarrow \{\text{Internal Forum Process}\}, \text{Trigger} = (\text{event}=>\{\text{state}\}, \text{operator} => \{\text{is}\}, \text{value} => \{\text{end}\}))$
- $\Phi_{111} \rightarrow XC (\text{External Entity} \Rightarrow \{\text{External Advisory Committee process}\}, \text{Trigger} = (\text{event}=>\{\text{state}\}, \text{operator} => \{\text{is}\}, \text{value} => \{\text{end}\}))$
- $\Phi_{112} \rightarrow XC (\text{External Entity} \Rightarrow \{\text{Business Case Approval process}\}, \text{Trigger} = (\text{event}=>\{\text{state}\}, \text{operator} => \{\text{is}\}, \text{value} => \{\text{end}\}))$

These newly identified constraint notations $\Phi_{100}$ to $\Phi_{112}$ will later contribute to building the full KAT Expression for the New Course Approval Process.

### 6.4.6 Representing Constraints on the Associations between Actions and Participants using KAT

Three types of constraints were identified that could affect the association between process participant and actions. These are i) a set of factors that are possessed by individual participants- (Individual characteristics), ii) constraints imposed based on
relationships that two participants may have – (Participant Comparison characteristics) and iii) external factors.

Here more KAT Definitions-19 and 20 are introduced to capture the above-mentioned constraints on the association between participants and process actions.

Individual characteristics refer to specific qualities possessed by a participant who is identified in the role. These were listed as the location, experience, skills or capabilities, availability, workload, history of performing process actions previously, preference, belonging to an organisational sub-structure and level of the organisational role of the participant.

Definition-19 shows a mechanism to capture these characteristics associated with a process participant into a KAT based guard element.

**Definition-19:** \( \varphi_x \rightarrow IC \) (role => {reference}, trigger => (characteristic => {location| experience| skills| availability| workload| past process action | org belonging | org level}, operator => {eq| gt| lt| el| eg | in | not in | yes | no | is | is not | associated}, value => {Figure}))

The important elements in this definition are: role- that identifies the process participant, characteristic- the particular aspect, which is captured into the constraint, operator- ways in which the association is measured, and value- an alpha numeric value that is used for comparison (see the example below). For example, consider the following constraint defined using Definition-9.

- \( \varphi_x \rightarrow IC \) (role => {manager}, trigger => (characteristic => {org belonging }, operator => {is}, value => {marketing}))

This denotes the constraint set to check whether the manager belong to an organisational unit named marketing.

Participant Comparison characteristics compare the two participants based on their role. These include belonging to same or different sub-structures, reporting and
delegation authorities and interpersonal relationships of two participants. Participant Comparison (PC) is presented in Definition-20.

**Definition-20**: $\phi_x \rightarrow PC(\text{role 1, role 2, (characteristic } => \{\text{organisational unit| personal relationships| delegate | report | order | level}, \text{ correlation } => \{\text{same | different | yes | no | higher | lower| before | after \} })$

According to Definition-20, participant comparison (PC) is described using role1, role2, characteristic, and correlation. Here certain characteristics of role1 and role2 are compared. The comparison characteristics could be their belonging to organisational units, interpersonal relationships, delegation, or reporting authorities, order of processing actions and level of the organisation.

There are two external factors affecting the association of actions to participants. These are time constraints and method of allocation of actions to participants deployed by the workflow management system.

As mentioned above, the method of allocation of actions to participants is depended on the particular implementation of a particular workflow management system. Therefore, method of allocation of actions is not considered as a constraint to be defined in the operational process at the semantic level. For imposing time constraints, it is possible to use previously given Definition-17.

### 6.4.6.1 Application of KAT Definitions-19 and 20 in OCAS

In OCAS related processes, individual characteristics are not taken into consideration when associating process actions with participants. In addition, external factors in the form of time constraints do not affect the association of process actions with participants. Therefore, there are no constraint notations defined based on Definitions-17 or 19.

Participant comparison is an important aspect in OCAS related process. For example, allocation of the approval task to a *Dean* for a course proposed by a particular *Academic*; depends on the belonging of the *Dean* and *Academic* to the same
organisational unit. Other participant comparison scenarios in OCAS, including the above, are listed in Table 6.10.

<table>
<thead>
<tr>
<th>Constraint notation</th>
<th>Description</th>
<th>Role 1</th>
<th>Role 2</th>
<th>Characteristic</th>
<th>Correlations</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi_{113}$ $\rightarrow$ PC (academic, Head of School, (characteristic =&gt; {organisational unit}, correlation =&gt; {same}))</td>
<td>Academic Head of school Organisational unit same</td>
<td>Academic and Head of School should belong to the same organisational unit</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\phi_{114}$ $\rightarrow$ PC (Head of School, Dean, (characteristic =&gt; {organisational unit}, correlation =&gt; {same}))</td>
<td>Head of school Dean Organisational unit same</td>
<td>Head of School and Dean should belong to the same organisational unit</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\phi_{115}$ $\rightarrow$ PC (academic, course officer, (characteristic =&gt; {organisational unit}, correlation =&gt; {same}))</td>
<td>Academic Course officer Organisational unit same</td>
<td>Academic and course officer should belong to the same organisational unit</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\phi_{116}$ $\rightarrow$ PC (Head of School, associate Dean- academic, (characteristic =&gt; {organisational unit}, correlation =&gt; {same}))</td>
<td>Head of school Associate Dean-Academic Organisational unit same</td>
<td>Head of School and Associate Dean Academic should belong to the same organisational unit</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\phi_{117}$ $\rightarrow$ PC (Head of School, Dean, (characteristic =&gt; {order}, correlation =&gt; {before}))</td>
<td>Head of school Dean Order in performing actions before</td>
<td>In the same process Head of School should always perform the actions before the Dean</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\phi_{118}$ $\rightarrow$ PC (pro-vice chancellor, Dean, (characteristic =&gt; {order}, correlation =&gt; {after}))</td>
<td>pro-vice chancellor Dean Order in performing actions after</td>
<td>In the same process pro-vice chancellor should always perform the actions after the Dean</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In here, six more phis ($\phi_{113}$ and $\phi_{118}$) are identified in relation to the OCAS New Course Approval Process. Constraints $\phi_{113}$ to $\phi_{116}$ defines the mechanism of identifying the correct participant, when there are more than one individual in the same role. For example, in UWS there are three Deans in three colleges. The relevant Dean for an appropriate process instance is chosen based on comparison between his/her association and Head of school’s association to the same organisational unit. The constraint notations $\phi_{117}$ and $\phi_{118}$ define the particular order that need to be persevered, due to reporting or delegation authorities.
6.4.7 Representing Constraints on the Associations between Actions and Business Object Elements using KAT

There were two types of constraints identified to affect the association between process elements and actions of the process. These are *ownership of data by an individual* and *specialisation of business objects*.

Object Constraints (OC) that affect the association between actions and business object is given in Definition-21.

**Definition-21:** \( \varphi_x \rightarrow OC \) (characteristic => \{ownership | specialisation\},

(identification => (object instance => \{attribute reference\}, correlation =>

{identification of the individual | specialisation details})))

The *characteristic* attribute first distinguishes between ownership and specialisation traits. *Object instance* gives a reference to identify a particular instance uniquely to which the ownership or specialisation is applicable. For example, this reference could be *loan application number* or *course code*. The correlation attribute identifies the particular individual to whom the loan application belongs to or the specialisation needs of the data.

In OCAS related processes, the constraints on the association between process actions and business object elements are not evident.

6.4.8 Representing Constraints on the Associations among Business Object Elements using KAT

In Chapter-5, it was identified the specialisation to be the only type of constraint that affect the associations within business object attributes. This aspect could be represented using Definition-21 above.

In OCAS related processes, specialisation constraints do not affect the associations among business object elements.
6.5 KAT Expression of the New Course Approval Process

Using the 118 constraint notations identified in the above discussions, the KAT Expression–9 introduced previously, can now be expanded. First, the composite guard elements \( C_1 \) to \( C_{32} \), \( C_m \) and \( C_n \) in KAT Expression-9 are described using the 118 constraint notations in Table 6.11.

Table 6.11- The expanded Composite Conditions in the New Course Approval Process

<table>
<thead>
<tr>
<th>Composite Phrase</th>
<th>Expanded Expression</th>
<th>Description</th>
</tr>
</thead>
</table>
| \( C_1 \)       | \( \varphi_1 \varphi_3 \varphi_6 \) | * \( \varphi_1 \) – academic is allowed to perform the action \( \alpha_1 \) AND  
* \( \varphi_3 \) – action \( \alpha_1 \) uses the interface definition given in this constraint AND  
* \( \varphi_6 \) – needs to preserve the integrity relationships identified in this constraint |
| \( C_2 \)       | \( \varphi_2 \varphi_3 \varphi_{11} \) | * \( \varphi_2 \) – Head of School (HOS) is obliged to perform the action \( \alpha_2 \) AND  
* \( \varphi_3 \) – action \( \alpha_2 \) uses the interface definition given in this constraint AND  
* \( \varphi_{11} \) – HOS and Academic belong to the same school |
| \( C_3 \)       | \( \varphi_3 \varphi_{42} \varphi_{114} \varphi_{117} \) | * \( \varphi_3 \) – academic is allowed to obliged the action \( \alpha_3 \) AND  
* \( \varphi_{42} \) – action \( \alpha_3 \) uses the interface definition given in this constraint AND  
* \( \varphi_{114} \) – Dean and HOS belong to the same college AND  
* \( \varphi_{117} \) – Checks whether a HOS has already performed the action |
| \( C_4 \)       | \( (\varphi_4 + \varphi_5) \varphi_{41} \varphi_{118} \) | * \( \varphi_4 \) – pro vice chancellor academic is obliged to perform the action \( \alpha_4 \) OR  
* \( \varphi_5 \) – secretary to vice chancellors advisory committee is permitted to perform the action \( \alpha_4 \) AND  
* \( \varphi_{41} \) – action \( \alpha_4 \) uses the interface definition given in this constraint AND  
* \( \varphi_{118} \) - Checks whether previous action was performed by a Dean |
| \( C_5 \)       | \( \varphi_6 \varphi_{42} \varphi_{71} \varphi_{96} \) | * \( \varphi_6 \) – project manager is obliged to perform the action \( \alpha_5 \) AND  
* \( \varphi_{42} \) – action \( \alpha_5 \) uses the interface definition given in this constraint AND  
* \( \varphi_{71} \) – course name information was filled in the previous action \( \alpha_1 \) AND  
* \( \varphi_{96} \) – needs to preserve the integrity relationships identified in this constraint |
| \( C_6 \)       | \( \varphi \varphi_{43} \varphi_{77} \varphi_{92} \varphi_{96} \) | * \( \varphi_7 \) – academic is obliged to perform the action \( \alpha_6 \) AND  
* \( \varphi_{43} \) – action \( \alpha_6 \) uses the interface definition given in this constraint AND  
* \( \varphi_{77} \) – related unit information was filled in the previous action \( \alpha_5 \) AND  
* \( \varphi_{92} \) – all units details are entered AND  
* \( \varphi_{96} \) – needs to preserve the integrity relationships identified in this constraint |
| \( C_7 \)       | \( \varphi_{44} \varphi_{113} \) | * \( \varphi_8 \) – Head of School is obliged to perform the action \( \alpha_7 \) AND  
* \( \varphi_{44} \) – action \( \alpha_7 \) uses the interface definition given in this constraint AND  
* \( \varphi_{113} \) – HOS and Academic belong to the same school |
| \( C_8 \)       | \( \varphi_7 \varphi_{45} \varphi_{78} \varphi_{108} \varphi_{109} \varphi_{110} \varphi_{111} \) | * \( \varphi_7 \) – Dean is obliged to perform the action \( \alpha_8 \) AND  
* \( \varphi_{43} \) – action \( \alpha_8 \) uses the interface definition given in this constraint AND  
* \( \varphi_{45} \) – course graduate attribute and unit information was previously entered in the action \( \alpha_2 \) AND  
* \( \varphi_{108} \) – Resource Assessment approval process is completed AND  
* \( \varphi_{109} \) – Course Code Entry Process is completed AND  
* \( \varphi_{110} \) – Internal Consultation Process is Completed AND  
* \( \varphi_{111} \) – External Advisory Committee Process is completed |
<table>
<thead>
<tr>
<th>Composite This</th>
<th>Expanded Expression</th>
<th>Description</th>
</tr>
</thead>
</table>
| C9             | (φ₁₀+φ₁₁)φ₆₆     | φ₁₀ – chair of college education committee is obliged to perform the action α₉ OR  
|                | φ₁₁₁ φ₁₁₄ φ₁₁₇    | φ₁₁ – secretary to college education committee is permitted to perform the action α₁₀ AND  
|                |                     | φ₄₆ – action α₉ uses the interface definition given in this constraint AND  
|                |                     | φ₁₁₁ – Business Case approval Process is completed AND  
|                |                     | φ₁₁₄ – Dean and HOS belong to the same college AND  
|                |                     | φ₁₁₇ – Checks whether HOS has already performed the action |
| C10            | (φ₁₂+φ₁₃)φ₄₇     | φ₁₂ – chair of course articulation and approval committee (CAAC) is obliged to perform the action α₁₀ OR  
|                |                     | φ₁₃ – secretary to CAAC is permitted to perform the action α₁₀ AND  
|                |                     | φ₁₂₇ – action α₁₀ uses the interface definition given in this constraint |
| C11            | (φ₁₄+φ₁₅)φ₁₆φ₉₆   | φ₁₄ – chair of academic senate is obliged to perform the action α₁¹ OR  
|                | (φ₁₆)φ₉₇φ₁₁₅      | φ₁₅ – secretary to academic senate is permitted to perform the action α₁₁ AND  
|                |                     | φ₁₄₈ – action α₁₁ uses the interface definition given in this constraint AND  
|                |                     | φ₁₆₆ – needs to preserve the integrity relationships identified in this constraint |
| C12            | φ₁₆φ₉₇φ₁₁₅        | φ₁₇ – course offer is obliged to perform the action α₁₂ AND  
|                |                     | φ₁₄₉ – action α₁₂ uses the interface definition given in this constraint AND  
|                |                     | φ₁₇₂ – course name information was previously entered in the action α₁ AND  
|                |                     | φ₁₅₃ – Course Officer and Academic belong to the same College |
| C13            | φ₁₇φ₃₀φ₇₃        | φ₁₇ – academic is obliged to perform the action α₁₃ AND  
|                |                     | φ₁₃₀ – action α₁₃ uses the interface definition given in this constraint AND  
|                |                     | φ₁₇₃ – course name information was previously entered in the action α₁ |
| C14            | φ₁₆φ₃₁ φ₇₆ φ₁₁₆  | φ₁₈ – associate Dean academic is obliged to perform the action α₁₄ AND  
|                |                     | φ₁₃₁ – action α₁₄ uses the interface definition given in this constraint AND  
|                |                     | φ₁₇₇ – needs to preserve the integrity relationships identified in this constraint AND  
|                |                     | φ₁₆₁₆ – Associate Dean and Academic belong to the same college |
| C15            | φ₁₉φ₅₂φ₇₄ φ₁₇    | φ₁₉ – project manager is obliged to perform the action α₁₅ AND  
|                |                     | φ₁₅₂ – action α₁₅ uses the interface definition given in this constraint AND  
|                |                     | φ₁₇₄ – basic course details were entered in the action α₁ AND  
|                |                     | φ₁₇₆ – needs to preserve the integrity relationships identified in this constraint |
| C16            | φ₂₀φ₃₃ φ₁₁₆        | φ₂₀ – associate Dean academic is obliged to perform the action α₁₆ AND  
|                |                     | φ₁₃₃ – action α₁₆ uses the interface definition given in this constraint AND  
|                |                     | φ₁₆₁₆ – Associate Dean and Academic belong to the same college |
| C17            | φ₂₁φ₅₄φ₇₅ φ₁₀₆   | φ₂₁ – project manager is obliged to perform the action α₁₇ AND  
|                |                     | φ₁₅₄ – action α₁₇ uses the interface definition given in this constraint AND  
|                |                     | φ₁₀₆ – course name information was previously entered in the action α₁ AND  
|                |                     | φ₁₀₆ – needs to preserve the integrity relationships identified in this constraint |
| C18            | φ₂₂φ₃₅φ₇₉        | φ₂₂ – director of marketing is obliged to perform the action α₁₈ AND  
|                |                     | φ₁₃₅ – action α₁₈ uses the interface definition given in this constraint AND  
|                |                     | φ₁₀₉ – career path information, reenrolment requirements and enrolment restrictions were previously entered in the action α₅ |
| C19            | φ₂₃φ₃₆φ₁₀₀      | φ₂₃ – director of information services division is obliged to perform the action α₁₉ AND  
|                |                     | φ₁₀₀ – action α₁₉ uses the interface definition given in this constraint AND  
|                |                     | φ₁₀₀ – required facilities information was previously entered in the action α₁₇ |
| C20            | φ₂₄φ₃₇φ₸₁       | φ₂₄ – director of education development unit is obliged to perform the action α₂₀ AND  
|                |                     | φ₁₃₇ – action α₂₀ uses the interface definition given in this constraint AND  
<p>|                |                     | φ₁₃₇ – training skills requirement was previously entered in the action α₅₇ |</p>
<table>
<thead>
<tr>
<th>Composite This</th>
<th>Expanded Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C21</td>
<td>φ₂₃φ₁₈φ₁₂</td>
<td>-φ₂₅ – librarian is obliged to perform the action α₂₁ AND &lt;br&gt;-φ₃₈ – action α₂₁ uses the interface definition given in this constraint AND &lt;br&gt;-φ₃₇ – requirements of library facilities were previously entered in the action α₁₇ AND</td>
</tr>
<tr>
<td></td>
<td>φ₂₆φ₃₉φ₃₅φ₄₄</td>
<td>-φ₂₆ – assistance registrar systems is obliged to perform the action α₂₂ AND &lt;br&gt;-φ₉₉ – action α₂₂ uses the interface definition given in this constraint AND &lt;br&gt;-φ₃₇ – enrolment requirements and enrolment restrictions were previously entered in the action α₁₇ AND &lt;br&gt;-φ₄₄ – course duration, course type and course offerings were previously entered in the action α₁</td>
</tr>
<tr>
<td></td>
<td>φ₂₇φ₆₀φ₆₅</td>
<td>-φ₂₇ – assistant registrar operations is obliged to perform the action α₂₃ AND &lt;br&gt;-φ₆₁ – action α₂₃ uses the interface definition given in this constraint AND &lt;br&gt;-φ₆₈ – enrolment requirements and enrolment restrictions were previously entered in the action α₁₇ AND &lt;br&gt;-φ₆₇ – delivery mode, type and location and total number of teaching hours were previously entered in the action α₃</td>
</tr>
<tr>
<td></td>
<td>φ₂₈φ₁₇φ₆₆φ₇₇</td>
<td>-φ₂₈ – college timetabling officer is obliged to perform the action α₂₄ AND &lt;br&gt;-φ₆₁ – action α₂₄ uses the interface definition given in this constraint AND &lt;br&gt;-φ₆₈ – required room and lab facilities information were previously entered in the action α₁₇ AND &lt;br&gt;-φ₇₇ – delivery mode, type and location and total number of teaching hours were previously entered in the action α₄</td>
</tr>
<tr>
<td></td>
<td>φ₂₉φ₃₂φ₳₈φ₁₀₀ φ₁₰₁ φ₁₀₂ φ₁₀₃ φ₁₀₄ φ₁₀₅ φ₁₀₆ φ₁₀₇</td>
<td>-φ₂₉ – senior planning architect is obliged to perform the action α₂₅ AND &lt;br&gt;-φ₆₂ – action α₂₅ uses the interface definition given in this constraint AND &lt;br&gt;-φ₆₈ – staffing and teaching facilities requirements details were previously entered in the action α₁₇ AND &lt;br&gt;-φ₁₀₀ – Library Sign-off process is completed AND &lt;br&gt;-φ₁₀₁ – ITD Sign-off process is completed AND &lt;br&gt;-φ₁₀₂ – EDU Sign-off process is completed AND &lt;br&gt;-φ₁₀₃ – Marketing sign-off process is completed AND &lt;br&gt;-φ₁₀₄ – Timetabling sign-off process is completed AND &lt;br&gt;-φ₁₀₅ – OAR-Systems sign-off process is completed AND &lt;br&gt;-φ₁₀₆ – OAR-Operations sign-off process is completed AND &lt;br&gt;-φ₁₀₇ – Capital Works Sign-off Process is completed</td>
</tr>
<tr>
<td></td>
<td>φ₃₀φ₆₃φ₁₁₄</td>
<td>-φ₃₀ – Dean is obliged to perform the action α₂₆ AND &lt;br&gt;-φ₆₃ – action α₂₆ uses the interface definition given in this constraint AND &lt;br&gt;-φ₁₁₄ – Dean and HOS belong to the same college</td>
</tr>
<tr>
<td></td>
<td>φ₃₁φ₆₄φ₇₆φ₉₆ φ₉₈ φ₉₉ φ₁₀₈</td>
<td>-φ₃₁ – project manager is obliged to perform the action α₂₇ AND &lt;br&gt;-φ₆₄ – action α₂₇ uses the interface definition given in this constraint AND &lt;br&gt;-φ₆₆ – basic course details were previously entered in the action α₁ AND &lt;br&gt;-φ₉₈ – needs to preserve the integrity relationships identified in this constraint AND &lt;br&gt;-φ₉₉ - Needs to maintain this computational constraint in calculating revenue AND &lt;br&gt;-φ₁₀₈ – Needs to maintain this computational constraint in calculating course cost AND &lt;br&gt;-φ₁₀₆ – Resource Assessment approval process is completed</td>
</tr>
<tr>
<td></td>
<td>φ₃₂φ₁₅</td>
<td>-φ₃₂ – business development officer is obliged to perform the action α₂₈ AND &lt;br&gt;-φ₁₅ – action α₂₈ uses the interface definition given in this constraint</td>
</tr>
<tr>
<td></td>
<td>φ₃₃φ₁₁₄</td>
<td>-φ₃₃ – Dean is obliged to perform the action α₂₉ AND &lt;br&gt;-φ₁₁₄ – Dean and HOS belong to the same college</td>
</tr>
<tr>
<td></td>
<td>φ₃₄φ₆₇φ₉₉φ₉₅ φ₁₁₈</td>
<td>-φ₃₄ – chair of VCAC is obliged to perform the action α₃₀ AND &lt;br&gt;-φ₆₇ – action α₃₀ uses the interface definition given in this constraint AND &lt;br&gt;-φ₉₉ – needs to preserve the integrity relationships identified in this constraint AND &lt;br&gt;-φ₉₅ – enrolment requirements and enrolment restrictions were previously entered in the action α₁₇ AND &lt;br&gt;-φ₁₁₈ – college timetabling officer is obliged to perform the action α₃₁ AND &lt;br&gt;-φ₁₁₄ – Dean and HOS belong to the same college</td>
</tr>
<tr>
<td>Composite Phis</td>
<td>Expanded Expression</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>( \varphi_{99} ) - course offering details were previously entered in the action ( \alpha_1 ), AND ( \varphi_{93} ) - A course is identified to be an ‘exception’ AND ( \varphi_{116} ) - checks whether the previous action was performed by a Dean</td>
<td></td>
</tr>
<tr>
<td>C_{31}</td>
<td>( \varphi_{35} ) ( \varphi_{68} ) ( \varphi_{90} ) ( \varphi_{94} ) ( \varphi_{36} ) - deputy vice chancellor development is obliged to perform the action ( \alpha_{31} ), AND ( \varphi_{68} ) - action ( \alpha_{31} ) uses the interface definition given in this constraint AND ( \varphi_{90} ) - course offering details were previously entered in the action ( \alpha_1 ), AND ( \varphi_{94} ) - course if planned to be offered in off-shore locations</td>
<td></td>
</tr>
<tr>
<td>C_{32}</td>
<td>( \varphi_{36} ) ( \varphi_{69} ) ( \varphi_{93} ) ( \varphi_{95} ) ( \varphi_{114} ) ( \varphi_{37} ) - Dean is obliged to perform the action ( \alpha_{32} ), AND ( \varphi_{69} ) - action ( \alpha_{32} ) uses the interface definition given in this constraint AND ( \varphi_{93} ) - course offering details were previously entered in the action ( \alpha_1 ), AND ( \varphi_{95} ) - Course is planned to be offered in onshore locations AND ( \varphi_{114} ) - Dean and HOS belong to the same college</td>
<td></td>
</tr>
<tr>
<td>C_{33}</td>
<td>( \varphi_{35} ) ( \varphi_{70} ) ( \varphi_{37} ) - any given time everyone is allowed just view the course data using action ( \alpha_{33} ), AND ( \varphi_{70} ) - action ( \alpha_{33} ) uses the interface definition given in this constraint</td>
<td></td>
</tr>
<tr>
<td>C_n</td>
<td>( \varphi_{100} ) ( \varphi_{101} ) ( \varphi_{102} ) ( \varphi_{103} ) ( \varphi_{104} ) ( \varphi_{105} ) ( \varphi_{106} ) ( \varphi_{107} ) All resource divisions have signed off</td>
<td></td>
</tr>
<tr>
<td>C_m</td>
<td>( \varphi_{119} ) Checks whether Course Code entry process is not complete OR Consultation Process is not Complete OR Resource Approval Process are is complete OR Business Case Approval Process is not complete</td>
<td></td>
</tr>
</tbody>
</table>

Using these expanded composite guard conditions given in Table 6.11, the full New Course Approval Process, complete with all alphas and phis, is given in KAT Expression 6.10.
E:= (φ₃φ₃₀φ₉₆)α₁ (φ₂φ₃⁹φ₁₁₄φ₁₁₇)α₂ ((φ₄+φ₅)φ₄₁φ₁₁₈)α₄ (φ₁₁₉)((φ₁₉φ₅₂φ₇₄φ₉₇)α₁₅ (φ₂₀φ₅₃φ₁₁₆)α₁₆)(1+
(φ₃₇φ₇₀)α₃₃) + ((φ₆φ₄₂φ₇₁φ₉₆)α₅(φ₇φ₄₃φ₇₇φ₉₂φ₉₆)α₆* (φ₈φ₄₄φ₁₁₃)α₇(φ₉φ₄₅φ₇₈φ₁₀₈φ₁₀₉φ₁₁₀φ₁₁₁)α₈((φ₁₀+φ₁₁)φ₄₆φ₁₁₁φ₁₁₄φ₁₁₇)α₉((φ₁₂+φ₁₃)φ₄₇)α₁₀ ((φ₁₄+φ₁₅)φ₄₈φ₉₆)α₁₁)(1+(φ₃₇φ₇₀)α₃₃) + ((φ₁₆φ₄₉φ₇₂φ₁₁₅)α₁₂ + (φ₁₇φ₅₀φ₇₃)α₁₃ (φ₁₈φ₅₁φ₉₇φ₁₁₆)α₁₄)(1+
(φ₃₇φ₇₀)α₃₃) + ((φ₁₉φ₅₂φ₇₄φ₉₇)α₁₅ (φ₂₀φ₅₃φ₁₁₆)α₁₆)(1+(φ₃₇φ₇₀)α₃₃) + ((φ₂₁φ₅₄φ₇₅φ₉₆)α₁₇ (φ₁₀₀φ₁₀₁φ₁₀₂φ₁₀₃φ₁₀₄φ₁₀₅φ₁₀₆φ₁₀₇)(
(φ₂₂φ₅₅φ₇₉)α₁₈ + (φ₂₃φ₅₆φ₈₀)α₁₉ + (φ₂₄φ₅₇φ₈₁)α₂₀ + (φ₂₅φ₅₈φ₈₂)α₂₁ + (φ₂₆φ₅₉φ₈₃φ₈₄)α₂₂ + (φ₂₇φ₆₀φ₈₅)α₂₃ + (φ₂₈φ₆₁φ₈₆φ₈₇)α₂₄ +
(φ₂₉φ₆₂φ₈₈φ₁₀₀φ₁₀₁φ₁₀₂φ₁₀₃φ₁₀₄φ₁₀₅φ₁₀₆φ₁₀₇)α₂₅ + ~((φ₂₉φ₆₂φ₈₈φ₁₀₀φ₁₀₁φ₁₀₂φ₁₀₃φ₁₀₄φ₁₀₅φ₁₀₆φ₁₀₇)(α₂₅ +1)))*
(φ₃₀φ₆₃φ₁₁₄)α₂₆)((φ₃₁φ₆₄φ₇₆φ₉₆φ₉₈φ₉₉φ₁₀₈)α₂₇ (φ₃₂φ₆₅)α₂₈(φ₃₃φ₆₆φ₁₁₄)α₂₉ ((φ₃₄φ₆₇φ₈₉φ₉₃φ₁₁₈)α₃₀ + (φ₃₅φ₆₈φ₉₀φ₉₄)α₃₁ +
(φ₃₆φ₆₉φ₉₁φ₉₅φ₁₁₄)α₃₂)) *(1+(φ₃₇φ₇₀)α₃₃)

**KAT Expression 6.10**- The New Course Approval Process of OCAS presented as a KAT Expression using alphas and phis
The KAT Expression 6.10 fully capture the entire set of CAD among process actions, participants and data object associated with the *New Course Approval Process* in OCAS. However, use of such linear expression is limited, without an implementation that allows this expression to be analysed for finding propagating impact. Therefore, the next section discusses a mechanism to implement KAT Expressions for analytical purposes.

### 6.6 Data Structure to Capture KAT Expressions

KAT Expressions cohesively and completely capture the CAD among process elements. This section investigates a suitable data structure that will allow capturing, manipulating, and analysing, these cohesive KAT Expressions.

Tree structures are considered efficient in representing complex and linear structures (Shave, 1975), similar to the ones in KAT expressions. In particular, binary trees are advocated to be used due to its regularity of the elements, which involves just three elements – left link, data, and right link. In addition, Shave (1975) demonstrates a mechanism to convert any general multi-branched tree structure into a binary tree structure, while preserving its structure.

There are five types of main flow constructs present in KAT expressions. These are: sequential flow of actions, iteration of an action, action guarded by a negated condition, conditional choice between actions and parallelism between actions. Figure 6.7 demonstrates the representation of these five types of flow control patterns in binary tree structures.
When creating and manipulating binary tree structures the following structural invariant needs to be maintained.

- There can be up to a maximum of two branches hanging from a particular node.
- Pphis and alphas always need to be in leaf nodes. In other words, pphis and alphas cannot be a parent node to another node.
- The guard element (a phi) of a particular action (an alpha) always needs to be in the left branch of the parent node.
- When the parent node holds a plus (+) operator as the node value, there has to be two children elements and they can be:
  - either two phis or two alphas - the combination of a phi and alpha cannot be hanging from a plus (+) operator parent node, or
  - a combination of any operators pluses (+), dots (.), stars (*) and negation (~).

Figure 6.7- Binary Tree representation of main flow control patterns in KAT
- When the parent node holds a dot (•) operator as the node value, there has to be two children elements and they can be a combination of any phis, alphas or any operator pluses (+), dots (•), stars (*), and negation (~).

- When the parent node holds a star (*) operator as the node value, there can be only one child node (as this is a unary operator) and the child element can be either an alpha, a plus (+) or a dot (•).

- When the parent node holds a negation (~) operator as the node value, there can be only one child node (as this is a unary operator) and the child element can be either a phi, a plus (+) or a dot (•).

The writing of a KAT expression into a binary-tree or recreation of KAT expressions from an existing binary-tree can be done using pre-order traversal mechanism. The pre-order sometimes known as symmetric order traversal mechanism reads the nodes in following order (Knuth, 1973):

- Read all nodes in the left side sub-tree of a node,
- Read the node, and
- Read all nodes in the right side sub-tree of the node.

This method guarantees that it will visit every element of the tree once and once only (Shave, 1975). In addition, it preserves the sequential ordering of actions when transforming to and from KAT expressions and tree structure.

Next, a sample KAT expression (KAT Expression 6.11) is taken to show the full view of a tree structure that would be created according to the set of invariants above. This KAT expression comprises of all possible flow constructs that were previously shown in Figure 6.8. KAT Expression 6.11 is for a hypothetical leave approval process, in which there are ten actions captured as alphas (α₁ to α₁₀) according to the KAT Definition-1. There are 32 guard elements, which appropriately defined using the other twenty definitions (Definition-2 to Definition-21).
KAT Expression 6.11 - Sample KAT expression of a hypothetical process

\[
\varphi_29((\varphi_{19} \varphi_7 \varphi_{25}) \alpha_1) + \sim \varphi_29(((\varphi_{19} \varphi_7 \varphi_{23}) \alpha_1)(\varphi_{m}(((\varphi_{21} \varphi_{12} \varphi_{32} \varphi_{27} \varphi_{30}) \alpha_3) ((\varphi_{22} \varphi_9 \varphi_{28} \varphi_{31}) \alpha_4) + (\varphi_{20} \varphi_{10} \varphi_{23}) \alpha_5) ((\varphi_{20} \varphi_{11} \varphi_{27} \varphi_{30}) \alpha_6)((\varphi_{20} \varphi_8 \varphi_{24}) \alpha_7) + ((\varphi_{22} \varphi_{13} \varphi_{23}) \alpha_8) ((\varphi_{22} \varphi_{14} \varphi_{28} \varphi_{31}) \alpha_9)))*((\varphi_{19} \varphi_7 \varphi_{24}) \alpha_{10})
\]

Figure 6.8 gives the binary tree structure that would represent the KAT Expression 6.11. This tree structure is created to preserve the set of invariants.

A tree structure (similar to Figure 6.9) gives the opportunity for doing various analyses on a process. Therefore, process represented in binary-tree structures similar to Figure 6.8 will be used in the **PECIA (Process Evolution and Change Impact Analysis) Model** in Chapter-8. In particular, when a process element changes, the propagating impact of the change, on other elements can be extracted using an appropriate algorithm, which can search a binary tree. An algorithm suitable for such searching KAT expressions would be discussed later in Chapter-8.
6.7 Chapter Summary

The main objective of this chapter was to find a mechanism to model complex correlations that exist among process elements. In this view, the author decided to use a formalism to capture CAD among process elements.

After analysing a number of formalisms including Petri-Nets, Process Algebra and KAT, it was decided to use KAT for modelling the complex correlations among process actions. The main reason of this selection of KAT was its bi-partite nature (having two types of elements named as alphas and phis) that allowed capturing CAD among process elements into linear expressions. Linear expressions, as oppose to graphical models, facilitate computer manipulation.

Based on the basic axioms of KAT, 21 definitions are introduced to capture CAD among process elements. These KAT definitions facilitate capturing all eight types of correlations (among process elements) previously identified in Chapter-5. Application of each of this KAT definition was demonstrated using the New Course Approval Process of OCAS. By applying KAT definitions to the New Course Approval Process, 33 phis (to represent process actions) and 118 alphas (to capture all the other associations) were introduced. Using these phis and alphas a KAT expression is developed to represent the New Course Approval Process (KAT Expression 6.10).

The final section of this chapter explored for a data structure that could be used to implement KAT expressions, in a form that it could be analysed. For this purpose, binary tree structures were considered. When modelling KAT expressions in a binary tree, the axioms of KAT are preserved using a set of invariants.

The main objective of capturing process element correlations into KAT expressions and representing them in binary-tree structure is to facilitate propagating impact analysis of process evolutions. The process elements can evolve in different ways and depending on the nature of the evolution, the extent of the ripple effect it creates varies. For this reason, it is imperative to comprehend the nature of process element evolutions for providing a complete solution for the issue of managing evolutions in web workflows. In this view, the next chapter carries out a detail investigation, to comprehend the nature of process evolutions.
Chapter-7

“One change always leaves the way open for the establishment of others.”
Niccolò Machiavelli (1469 – 1527)

7 Nature of Process Element Evolution

7.1 Chapter overview

Elements in business processes (actions, participants, and business object) have different categories of associations among them. These associations among process elements were extensively discussed in Chapter-5. Then Chapter-6 provided a mechanism to capture these associations cohesively and completely into linear expressions using Kleene Algebra with Tests (KAT). The objective, of capturing processes into linear KAT expressions, was to analyse them to locate the propagating impact of element changes.

The propagating impact does not exclusively depend on the associations among process elements. The other factor that contributes to propagating impact is the nature of process element evolution. With this view, this chapter discusses the primitives, semantics, and dynamics of process element evolution.

The alterations that take place in a process element can be classified into three types as modifications, additions and deletions, which are referred to as primitives of process element evolution. These primitives can be applied to each element, resulting in a combinatorial set: (i) action additions, (ii) action modifications, (iii) action deletion, (iv) participant additions, (v) participant deletions, (vi) participant modifications, (vii) business object additions, (viii) business object deletions, and (ix) business object modifications.

When evolution primitives are applied to process elements (actions, participants and business object) these elements have varying behaviours; due to the distinct nature of attributes that characterises each element. For instance, action
modification does not have the same semantics as participant modification. Action modification may mean changing its existing sequential flow into a parallel flow; and participant modification may signify changes in the responsibilities associated with the organisational role that identifies the participant. These diverse types of evolutions in process elements are termed as semantics of process element evolution.

The impact of an element evolution onto other elements, depend on the evolution semantics associated with it. For instance, consider the two action modifications: (a) changing an action association with a participant, and (b) changing the order in which the same action appears in the process. Modification of ‘action associations with a participant’ results in allocating the action to another participant; and has less or no impact on other actions. However, the modification-‘change ordering of action’ may have significant impact on other actions that are depending on the former action to enter data for routing. This aspect of having varying types of impacts on other actions is referred to as dynamics of process element evolution.

Based on these primitives, semantics and dynamics of process element evolution, the changes that need to take place in other process elements can be prioritised. For instance, removal of a participant (identified in an organisational role) from the organisational structure (for example, for strategic reasons role named ‘line supervisor’ is identified as redundant) has an immediate impact on the actions that are performed by the removed participant. However, this same change may have less or no impact to other actions.

This research classifies four categories of impacts as direct, indirect, and secondary and Non-cautionary (DISN). These DISN terms suggest the severity of impacts. For instance, direct impacts have more damages (and cannot be passed without corrections) and cautionary impacts have least destruction to the process. Indirect impact means actions that cannot be performed because of direct impacts. Secondary impacts signify actions that can be performed, but cannot merge with the main process due to direct impacts. This chapter extends its discussion to describe and exemplify finding these DISN-impacts, using KAT expressions of a process.

The remainder of this chapter is organised as follows. Firstly, an investigation is carried out to find the attributes that characterise each business process element. Then the semantics of process evolution is extensively discussed under each process element. Followed by this discussion, the dynamics of process element evolution are
presented. The final section uses the findings of evolution semantics and dynamics, and KAT expressions, to exemplify locating of DISN-impacts.

7.2 Characterisation of Process Elements

A number of attributes characterise each type of business process elements—actions, participants and business object. By analysing previous work in business process automation—BPA (Russell et al., 2005b; Kappel et al., 2000; Chiu et al., 1999; Joeris & Herzog, 1998; Kappel et al., 1995), and in the area of business rules (SEEC, 2003; Hay et al., 1997; Gottesdiener, 1997; Halle, 1996) attributes that characterise process elements can be categorised as follows:

- **Identification attributes**: These refer to names or IDs that individually recognised process elements among many. For instance, action names (approve course) or name of the role of the participant (Head of School) are some identification attributes.

- **Categorisation attributes**: These attributes categorise process elements into groups. Process elements can by categorised in multiple ways. Following are some examples of categorisations of actions, participants, and business object data.
  
  - Process actions can be categorised as manual, partially automated or fully automated (van der Aalst et al., 2003b). Another type of action categorisation is value adding, gate-keeping or communicative actions (this type of categorisation was described in detail in Chapter-5).
  - Process participants can be categorised as humans, machines or computer programs (Russell et al., 2005b). Another categorisation of participants is their internal or external nature to an organisation.
  - Business data elements have categorisations such as production or control data; internal or external data; core or non-mandatory information; and manual or digitised data.

The categorisation of elements are not limited to the above examples, but can be extended to other various categories as required by the business process.

- **Descriptive attributes**: These attributes further describe process elements, based on some features pertaining to it, but without associating other elements. For example, process participants may have responsibilities or capabilities.
associated with them, and object data attributes may have length, format and data type features. These attributes do not uniquely identify or categorise a process element, but extend the characterisation of it without associating it to another.

- **Associative attributes**: As the name suggests associative attributes define a process element based on its relationships with another process element. The action *approve course* is carried out by *Dean*, and is an example of an action associated with a participant.

Table 6.1 lists the various characterisations attributes and their applicability to each process elements.

<table>
<thead>
<tr>
<th>PROCESS ELEMENT TYPE OF ATTRIBUTE</th>
<th>PROCESS ELEMENT ACTIONS</th>
<th>PROCESS ELEMENT CUSTOMERS</th>
<th>PROCESS ELEMENT OBJECTS DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Identification attributes</strong></td>
<td>Action name</td>
<td>Organisational role of the participant</td>
<td>Types of categorisations of data elements would be;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- manual (hard copy documents/forms) or digital format</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- document based or database oriented</td>
</tr>
<tr>
<td><strong>Categorisation attributes</strong></td>
<td>There could be different types of action categorisations such as;</td>
<td>Different types of participant categorisations would be similar to;</td>
<td>- field type</td>
</tr>
<tr>
<td></td>
<td>- Manual, partially automated, fully-automated</td>
<td>- human, machines or computer programs</td>
<td>- field length</td>
</tr>
<tr>
<td></td>
<td>- Value adding, gate keeping, communicative</td>
<td>- external or internal to the organisation</td>
<td>- field format</td>
</tr>
<tr>
<td><strong>Descriptive attributes</strong></td>
<td>Action descriptions or any other information that further describes the action, without linking it with another action, participant or object data</td>
<td>- responsibilities or duties</td>
<td>Shows associations with other object data elements or another process element</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- capabilities, skills or experience that needs possessed by the participant</td>
<td>- integrity relationships</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- belongs to an organisation unit or group</td>
<td>- computational relationships</td>
</tr>
<tr>
<td><strong>Associative attributes</strong></td>
<td>Shows the associations that actions have with other elements</td>
<td>Shows particular associations that participants may have with other elements</td>
<td>Shows associations with other object data elements or another process element</td>
</tr>
<tr>
<td></td>
<td>- Action is in sequence with another action</td>
<td>- participant may own certain object data instances</td>
<td>- integrity relationships</td>
</tr>
<tr>
<td></td>
<td>- Action is obliged/ permitted/ forbidden to be performed by a participant</td>
<td>- reporting or delegation authorities in relation to another participant</td>
<td>- computational relationships</td>
</tr>
<tr>
<td></td>
<td>- action has data visibility associated with it</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The characterisation of elements, based on its association with other elements, creates a ‘chicken and egg’ situation, which raises the question similar to,
“Do the actions characterise participants or vice versa?” From the processes point of view, the most important elements are the process actions. In other words, the existence of other elements (in a particular process), such as participants or object data, are not acceptable if these are not associated with an action. For example, when characterising the action ‘recommend course’ it will identify ‘Dean’ as the role to perform the action; as opposed to creating an action, just because a participant is there.

These associative attributes is the main cause for impact propagation in process element evolution. Therefore, it is required to analyse the evolutions that can take place closely in associative attributes. Conversely, there is less (or sometimes no) impact to other elements when categorisation or descriptive attributes are changed. The impact of changes to identification attributes depend on the implementation architecture of a workflow management system (WfMS). For example, an elegant implementation of a WfMS would facilitate adapting to action name, participant role or data attribute name changes, without creating runtime errors. In this research, the author assumes that web-based WfMSs under study have this minimum flexibility.

Based on above process characterisation attributes, the next section details the semantics of process evolution.

7.3 Semantics of Process Element Evolution

The semantics of process element evolution refers to comprehending the behaviour of process elements (actions, participants and business object), when different types of changes – additions, modifications and deletions take place. These types of changes are referred to as primitives of evolution.

Primitives of evolution are previously studied by many, in different other areas. One of the first systematic studies comes from Banerjee et al. (1987), in which the semantics of object oriented databases schema evolution is explained. They identify the primitives of evolution to be additions, deletions and modifications. Later others such as Casati et al. (1998) have continued to use these primitives in workflow research.

From the workflow evolution perspective van der Aalst and Jablonski (2000) have given another classification to define the changes that can take place in process
elements. This categorisation include, *extensions* – similar to additions discussed above, *reductions* – similar to deletions, and *re-linking* – which is defined as both extension and reduction taking place. These classifications are also based on Banerjee et al. (1987) primitives. Therefore, this research considers *additions*, *deletions*, and *modifications* to be the primitives of process element evolution.

Remainder of this section details the *semantics of process element evolution*, under three headings: *Action evolution semantics*, *Participant evolution semantics*, and *Business object evolution semantics*.

### 7.3.1 Action Evolution Semantics

Evolution of actions is one of the highly discussed aspects in previous research related to workflow evolution. Casati (1998) refers to the action evolution in general to be *flow primitives evolution* in relation to changing the ordering of tasks. van der Aalst et al. (1999) identifies the same, under the topic *task dimension of evolution*.

In the area of workflow evolution, Casati et al. (1998), Liu et al. (1998) and Sadiq et al. (2000a) introduce the importance of having a modification language to deal with evolution. Syntax of such modification language is dictated by the workflow modelling language and the implementation of WfMS. The focus of this research is at a higher-level rather than at one particular process automation mechanism. Therefore, this work does not attempt to develop a series of process modification functions. Nevertheless, here a set of parameters are identified, in relation to evolution, which can be passed into a particular modification function developed in another implementation.

Here a simple symbolic representation that is used to explain existing and new flow constructs. Consider where actions A,B,C and D have changed from a sequential ordering to a parallel branching. In this case, the existing and new orderings are as follows:

- Existing ordering (A, B, C, D)
- New parallel branching (A,B,D) and (A,C,D)

In the same example, if the new ordering is from a sequential to a conditional choice, in which the choice is denoted by \((r=q)\) and \((r>q)\), existing and new ordering are given as follows:
- Existing ordering (A, B, C, D)
- New parallel branching (A,(r=q)B,D) and (A,(r>q)C,D)

Table 6.2 details three types of action evolutions namely: (i) addition of actions, (ii) deletion of actions, and (iii) modification of actions. Against each of these action evolution types, sub-categories of evolution are discussed, in the second column of the table. These sub-categories further analyse each evolution type based on various associations that process actions have with other elements. The third column explains the evolution semantics using the above-described syntactical representation. Finally, the fourth column shows the before and after view of the evolution in a diagrammatic form.

**Table 7.2- Semantics of Process Action Evolution**

<table>
<thead>
<tr>
<th>Evolution Primitive</th>
<th>Sub-categories of action evolution</th>
<th>Parameters that required to be specified</th>
<th>Before and After view</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addition</td>
<td>Into a sequence</td>
<td>i) name of the added action – X</td>
<td>Before</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ii) existing ordering of actions (A,B)</td>
<td>A → B</td>
</tr>
<tr>
<td></td>
<td></td>
<td>iii) new ordering of action (A,X,B)</td>
<td>After</td>
</tr>
<tr>
<td>As a parallel action</td>
<td></td>
<td>i) name of the newly added action- X</td>
<td>Before</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ii) existing branching (A, B, D) and (A,C,D)</td>
<td>A → B, C → D</td>
</tr>
<tr>
<td></td>
<td></td>
<td>iii) new branching (A,B,D), (A,C,D) and (A,X,D)</td>
<td>After</td>
</tr>
<tr>
<td>As a conditional-choice action</td>
<td></td>
<td>i) name of the added action - X</td>
<td>Before</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ii) existing ordering of actions (A, (r=c)C, D) and (A, (r=b)B,D)</td>
<td>A → B, C → D</td>
</tr>
<tr>
<td></td>
<td></td>
<td>iii) New ordering of actions (A, (r=c)C, D), (A, (r=b)B, D) and (A, (r=x)X, D)</td>
<td>After</td>
</tr>
<tr>
<td>Deletion</td>
<td>From a sequence</td>
<td>i) name of the deleted action – X</td>
<td>Before</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ii) existing ordering (A,X, B)</td>
<td>A → X → B</td>
</tr>
<tr>
<td></td>
<td></td>
<td>iii) new ordering (A,B)</td>
<td>After</td>
</tr>
<tr>
<td>Evolution Primitive</td>
<td>Sub-categories of action evolution</td>
<td>Parameters that required to be specified</td>
<td>Before and After view</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------</td>
<td>-----------------------------------------</td>
<td>----------------------</td>
</tr>
</tbody>
</table>
| From a parallel set of action | i) name of the deleted action - X  
ii) Existing ordering of actions (A,B,D), (A,C,D) and (A,X,D)  
iii) new ordering (A,B,D), (A,C,D) | Before and After view diagram |
| Delete action from a conditional choice | i) name of the deleted action - X  
ii) existing ordering (A, (r=c)C, D), (A, (r=b)B, D) and (A, (r=x)X, D)  
iii) new ordering (A, (r=c)C, D) and (A, (r=b)B, D) | Before and After view diagram |
| Modifications | Change the name of the action | i) previous name- A and new name - X |  |
| Change the participant (associations obligations, permissions or forbiddances remain as it is) | i) name of the action - X and existing participant -P Association(P,X)  
ii) new participant –R Association(R,X) |  |
| Change the associations with a participant (participant continues to be the same) | i) Action X with participant P  
ii) the existing association and new association as one of the following  
(obligation, permission)  
(permission, obligation)  
(obligation, forbiddance)  
(forbiddance, obligation)  
(forbiddance, permission)  
(permission, forbiddance) |  |
| Change the action interface by changing the visibility aspect of data items | i) name of the action and related object data element (X, data_element)  
ii) the existing visibility trait and the new visibility trait, as one of the following:  
(Edit, Hidden)  
(Hidden, Edit)  
(Edit, View)  
(View, Edit)  
(View, Hidden)  
(Hidden, View) |  |
<p>| Changing the order of actions from | i) existing ordering (A, B, C, D) | Before and After view diagram |  |</p>
<table>
<thead>
<tr>
<th>Evolution Primitive</th>
<th>Sub-categories of action evolution</th>
<th>Parameters that required to be specified</th>
<th>Before and After view</th>
</tr>
</thead>
</table>
| - Sequence to parallel | ii) new parallel branching (A,B,D) and (A,C,D) | | ![Diagram 1](A→B→C→D)  
After: ![Diagram 2](A→B→C→D) |
| Changing the order of actions from Parallel to sequence | i) existing ordering (A,B,D) and (A,C,D)  
ii) new sequential ordering (A,B, C, D) | | ![Diagram 3](A→B→C→D)  
Before: ![Diagram 4](A→B→C→D) |
| Changing the order of actions from - Sequence to conditional split | i) existing ordering (A, B, C, D)  
ii) condition and values associated with new conditional branching of actions (A, (r=b)B, D) and (A, (r=c)C, D) | | ![Diagram 5](A→B→C→D)  
Before: ![Diagram 6](A→B→C→D) |
| Changing the order of actions from - Conditional split to sequence | i) existing branching (A, (r=b)B, D) and (A, (r=c)C, D)  
ii) condition that get removed (r)  
iii) new sequential ordering of actions (A,B,C,D) | | ![Diagram 7](A→B→C→D)  
Before: ![Diagram 8](A→B→C→D) |
| Changing the order of actions from - Parallel to conditional split | i) existing ordering (A,B,D, E) and (A,C,D,E)  
ii) conditional values and new conditional branching (A, (r=c)C, E) and (A, (r=b)B, D, E) | | ![Diagram 9](A→B→C→D)  
Before: ![Diagram 10](A→B→C→D) |
| Changing the order of actions from - Conditional split to parallel | i) existing ordering (A, (r=c)C, E) and (A, (r=b)B, D, E)  
ii) condition that get removed - r  
iii) new parallel branching – (A,B,D, E) and (A,C,D,E) | | ![Diagram 11](A→B→C→D)  
Before: ![Diagram 12](A→B→C→D) |
This section presented the *semantics of process action evolution*. The types of evolutions associated with process actions can be broadly categorised into three as follows:

- Addition of actions - A(a)
- Deletion of actions - D(a)
- Modification of actions - M(a)

Later in the discussion, the symbolic notations for A(a), D(a) and M(a) are used to refer to additions, deletions and modifications of actions respectively.

### 7.3.2 Participant Evolution Semantics

The evolutionary aspect of process participants is discussed in relation to three evolutionary primitives - additions, deletions and modifications, similar to above.

#### 7.3.2.1 Additions and Deletions of Participants

The addition of a process participant necessarily means adding a role to the organisational structure. This type of change has less or no impact, unless that role is associated with a process action. The association of process participants was discussed under action evolution in the previous section.

Deletion of a role from an organisational structure may have significant impact on the process, especially if that role has some associations with process actions. In this situation, it is imperative to flag the need for changes in the related actions.

Participants have three different types of associations with an action as *obligations, permissions, and forbiddances* (Chapter-5 discussed these associations in detail). If the association between action and participant is a *forbiddance*, then there is no impact on the action, other than removing this constraint from process model, as it is no longer valid. If the association between action and participant is a *permission*, still there is no significant impact on the process. However, the process definition is semantically incorrect, as the role does not exist in the organisation. Therefore, the process models are required to change accordingly. In the case where participant has an *obligation* to perform an action, it creates operational issues. Such
actions are required to be identified correctly and assigned to other roles, to eliminate the impact.

7.3.2.2 Modification of Participants

Modification of participants can taken place from different aspects. The following list represents those changes.

- **Change a name of the existing role**: This assumes only changes in the role name (for example, from *Head of School* to *Head of the Faculty*). This change is purely a change into the identification attribute of the participant and does not involve any changes to the associations with actions. These types of changes are expected to be handled by the WfMS as appropriately. However, process models need to be changed accordingly to reflect the role name change.

- **Change role’s association with an organisational unit**: Role of a participant (for example, manager) is generally associated with organisational units (for example, marketing division). Sometimes due to strategic reasons, organisational units are amalgamated into a single unit, or sub-divided into multiple units. In such situations, association of roles with organisational units change. These changes create issues with the constraints (identified as phis in KAT expressions – discussed in Chapter-6) that capture such associations. Therefore, it is required to identify such constraints and change them accordingly, to reflect the real-life evolution scenarios.

- **Change the responsibilities associated with a role**: Organisational roles have various responsibilities associated with them. Sometimes roles are assigned more duties and responsibilities; or existing responsibilities may be divided among two or more roles. Changes similar to the above may result in either merging two or more actions into a single action or splitting an action/s into several actions. The merging of actions is similar to both additions and deletions taking place at the same time, which is referred to as *re-linking* by van der Aalst and Jablonski (2000).

- **Change the reporting and delegation structure of the role**: Every role in an organisation is required to report to some other role, board, committee, or shareholders of the organisation. When positioning actions and associating them to roles, this reporting structure is taken into consideration. For
example, in the courses approval process the Head of School reports to Dean. Therefore, when both Head of School and Dean perform actions in a process, the Dean’s action is usually after the Head of school’s action. Changes to this reporting structure would certainly affect the process in relation to action organisation, which was discussed in the previous section.

This section discussed the semantics of process participant evolution from the organisational point of view. Broadly, three types of participant evolutions were discussed as follows;

- Addition of a participant - \( A(p) \)
- Deletion of a participant - \( D(p) \)
- Modification of a participant - \( M(p) \)

These kinds of participant evolutions occur due to strategic and operational needs of the organisation, which are beyond the control of the process. However, when processes are designed certain characteristics associated with process participants are taken into consideration. When changes take place based on these characteristics, it is likely that it will have propagating impact on the process. This aspect is discussed later in this chapter under the dynamics of evolution.

### 7.3.3 Business Data Object Evolution Semantics

Business object of a process can evolve and change beyond the control of the process. For instance, course object of new courses approval process (introduced as the case study in Chapter-4), is changed from time-to-time to cater for various business needs.

The semantics of business data object evolution is considerably different from previously studied two process elements - actions and participants. This is due to the diversity of attributes that characterise business data object elements.

In relation to business processes, evolution of business data object and its impact on process is inadequately researched. Therefore, semantics of business object evolution is discussed below, in relation to these three evolution primitives – additions, deletions and modifications.
### 7.3.3.1 Addition and Deletion of Business Data Object Elements

This refers to adding more fields into the database or the document that implements the process object. Generally, such additions are done in order to capture more information or to support process routing.

When fields are added, it has an impact on the visibility of data in action interfaces (data visibility was discussed as one type of association between business object and process actions). Therefore, the action visibility needs to be amended accordingly in implementation artefacts of the workflow system, via appropriate models.

The other common change that takes place in business data objects is the removal of unnecessary fields. Fields may be removed, as new ones are introduced or existing ones are modified. For instance, in a leave approval process, *number of leave days* field could be removed as the two new fields *leave start date* and *leave end date* are introduced. Such deletions would have an effect on process action interfaces and routing. Therefore, these deletes need to be handled with care so that it does not break the consistency of the process flow.

### 7.3.3.2 Modification of Business Data Object Elements

Modifications of business data object fields have many forms. Below each type of change is discussed in detail. These types of changes are based on the concepts of schema evolution in object oriented databases (Banerjee et al., 1987).

- **Changes to the element name:** Changing an existing field name could be one of the simpler changes that can take place in a database field. It may have significant impact on the process, particularly if these fields are referred to get data to make routing decisions, which is also referred to as data based routing (Russell et al., 2005a). As mentioned in the previous section 7.2, a field name is an *identification attribute*. Therefore, it is expected the WfMSs ability to manage such changes.

- **Changes to element integrity relationships:** Fields in a database have relationships with other fields or tables. Such relationships are implemented to preserve the integrity and consistency of a data set. In relational databases, this would be in the form of foreign key relationships. In a true object oriented database, such relationships would be via methods. In a document-based system, these relationships would be simply via hyperlinks.
Irrespective of the integrity relationship mechanism, changes to such relationships could create issues in relation to visibility of fields in action interfaces. Therefore, for situations where strict integrity relationships need to be maintained, it needs to be captured into a constraint (a phi element in KAT introduced in Chapter-6).

• **Changes to the element computational relationships:** Computational relationships refer to field values that are calculated using other elements. Changes in such computational relationships have similar effect to integrity relationships discussed earlier. Therefore, the important computational relationships need to be captured into a constraint in the process model.

• **Changes to the type of the element:** The change of data field types is another change that can take place in a database. For example, integer fields can be converted to floats allowing decimals to be entered. Such changes are simple, yet have an impact on the action interfaces, particularly in the case of forms, where it requires validations of data entered by the user.

• **Changes to the length of the element:** Field length changes are similar to type changes. This refers to a simple change of extending or shortening the field length of the element in the database. This is an impact on the action interfaces, particularly form based interfaces.

• **Changes to the format of the element:** Certain fields may have strict formatting imposed. For example, product codes or employee numbers would adhere to certain organisational standards and could be a strict format. Thus, changes to such formatting, may require changes in validations at the action interfaces.

• **Changes to the compulsoriness of the element:** In a database, certain fields can be made compulsory for data entry. Most databases management systems support monitoring mandatory nature of a field. Similar to changes discussed above, the changes to compulsoriness of data fields creates validation issues at the action interfaces.
This section, introduces three more evolutionary aspects of a process as follows;

- Addition of a business data object elements - A(d)
- Deletion of a business data object elements - D(d)
- Modification of a business data object elements - M(d)

This section discusses nine types of evolutions that can take place in processes as: Addition of actions-A(a), Deletion of actions-D(a), Modification of actions-M(a), Addition of a participant-A(p), Deletion of a participant-D(p), Modification of a participant-M(p), Addition of a business data object elements-A(d), Deletion of a business data object elements-D(d), and Modification of a business data object elements-M(d). The semantics of each of the above evolutions were discussed. The next section, details the propagating impact of above evolutions under the term *dynamics of process evolution*.

### 7.4 Dynamics of Process Element Evolution

Process elements do not exist in isolation and are woven together to achieve business goals. Therefore, changes in one element have propagating impact on other elements. This propagating impact is referred to as dynamics of process evolution. In this section, the dynamics of process evolutions are discussed in relation to nine types of evolutions identified in the previous discussion.

The propagating impact refers to one of the above types of change causing an immediate change type. For example, consider the deletion of a participant denoted by D(p). As a consequence of such a change, it is compulsory to modify process actions to assign those actions to another participant, which is denoted by M(a). This dynamic of deletion of process participant and its impact can be graphically denoted as shown in Figure 7.1.

![Figure 7.1- Compulsory Propagating Impact of Deletion of a Process Participant](image_url)
In some situations, there could be propagating changes that may not be compulsory to take place, but are possible based on the process. For instance, addition of a process participant A(p), may not always result in having to add a new action A(a), but it could be possible for it to happen. Thus such non-compulsory impacts are denoted using dotted arrow as shown in Figure 7.2.

![Figure 7.2- Non-Compulsory Propagating Impact of Addition of a Process Participant](image)

This compulsory and non-compulsory impact of process element changes is referred to as dynamics of process evolutions. Next, a series of diagrams are presented using the above-mentioned notation in order to detail the dynamics of element evolution.

Figure 7.3 shows the compulsory and non-compulsory changes propagated because of addition, deletion, and modification of actions.

![Figure 7.3- Dynamics of Addition, Deletion and Modification of Process Actions](image)

Figure 7.4 shows the compulsory and non-compulsory propagating changes in relation to addition, deletion, and modification of participants.
Figure 7.5 shows the compulsory and non-compulsory propagating impact of addition, deletion, and modification of elements of the business data object associated with a process.

7.4.1 Example of Dynamics of Process Evolution

In order to describe the above presented three models of dynamics of process evolutions (Figures 6.3, 6.4 and 6.5) a sample scenario taken from the OCAS New Course Approval Process is used.

Figure 7.6 shows the course notification of intention (CNI) and business case (BC) approval processes, as it occur in the New Course Approval Process in OCAS.
In the new courses approval process, initial intention for developing a new course is required to be approved by the Pro-Vice Chancellor-Academic (PVC-A Approval of CNI bubble in the above diagram). This initial approval is referred to as Course Notification of Intention (CNI), which collects a specific set of information in relation to the course that is proposed to be developed. In CNI approval, there are four major states. First, the project manager, who is also called a proposer, creates a CNI document. This creation of CNI document is the value adding action in the CNI approval process. Then the CNI is submitted to the Head of School as the gatekeeping function at the school level. At the college level Dean recommends the CNI, before the final approval by PVC-A.

Once the CNI is approved by PVC-A, a Business Case (BC), is created for the course among many other sub-processes, which are not considered in Figure 7.6 above. In the BC approval process, first the BC is prepared by the project manager. Then this is signed-off along with an independent assessment by the Office of Business Development (OBD). As the sponsor of the course, the Dean assesses the BC, before the final approval. The final approval of the BC could be given either by the PVC-A, Deputy-Vice Chancellor-development and International (DVC-D&I) or Dean. This is decided upon, some data associated with the course. If the OBD specifies a course to be an ‘exception’ (example requires building new labs with significant costs), then it is required to be approved by PVC-A. If the course is an ‘off-shore course’ (offered at overseas locations) and is not marked to be an exception, then it is approved by DVC-D&I. If the course is a ‘domestic course’ and not marked as an exception, it will get final approval by the Dean.
Table 7.3 discusses the dynamics of process evolution presented in Figures 6.4, 6.5 and 6.6 with some examples drawn from the process explained in Figure 6.7.

<table>
<thead>
<tr>
<th>Initiating Evolution</th>
<th>Propagating Change</th>
<th>Compulsory or non-compulsory</th>
<th>Description and/or Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>A(a)</td>
<td>M(a)</td>
<td>Compulsory</td>
<td>When an action is added to a process, it enforces to change the order of existing actions as it was shown in Table 7.2 under addition of actions.</td>
</tr>
</tbody>
</table>
|                      | D(a)               | Non-compulsory                | - A new action ‘College Submission of CNI’ is added to the process which is to be performed by the Dean of the college.  
- This new action has the same capacity for it to be the gate keeping function at the college level.  
- This may result in deleting the existing action of ‘Recommend CNI’ |
|                      | A(p)               | Non-compulsory                | - In relation to BC approval a new action ‘Enter Fee Information’ is added.  
- Since the fee information is managed separately by a fee officer, it may require the ‘fee officer’ to be included as a new participant to the process |
|                      | M(p)               | Non-compulsory                | - A new action ‘Calculate course costs’ is included to the BC approval process.  
- This particular action may not be a responsibility that is included in the job description of any of the roles in the organisation.  
- Therefore, this new addition of action may result in having to change the responsibilities of an existing participant to include the task. Hence, a modification of process participant takes place due to the new action. |
|                      | A(d)               | Non-compulsory                | - When new actions are added to a process, the existing data set of the object may not be able to support it fully.  
- In order to perform this action, new action ‘enter fee information’, it may be vital to have some information about the course, such as level of the staff personnel teaching the course, that were not considered previously.  
- This results in adding the new data elements to capture the staff information to assist the proper function of the newly added action. |
|                      | D(a)               | Compulsory                    | When an action is deleted, it essentially breaks the flow of the existing actions. Thus, it is compulsory for some of the exiting actions to be modified accordingly. |
|                      | D(p)               | Non-Compulsory                | - Consider removing the action ‘Sign-off BC with independent assessment’ from the BC approval process.  
- The Officer in Office of Business Development (OBD) was only involved, in this action in this process. Therefore, that role may not be further required in this process.  
- From the point of this process, that role could be removed. |
<table>
<thead>
<tr>
<th>Initiating evolution</th>
<th>Propagating Change</th>
<th>Compulsory or non-compulsory</th>
<th>Description and/or Example</th>
</tr>
</thead>
</table>
| **M(p)**            | Non-Compulsory      | - Removal of a role due to an action removal mentioned earlier is an extreme situation.  
                    |                     |   - However, most likely scenario would be modification of participant by changing the responsibilities associated with the roles.  
                    |                     |   - Thus, a deletion of an action may result in modification of participants. |
| **D(d)**            | Non-compulsory      | - When an action is deleted from the process, certain data that was particularly associated with that action may no longer required.  
                    |                     |   - Thus it may result in deleting some data elements from the business object of the process. |
| **M(a)**            | Non-compulsory      | - An action modification may refer to changing the existing ordering to a different one. For example from a sequence to a parallel.  
                    | (Depicted Figure 7.3) |   - This kind of change have affect on some of the other actions. |
| **A(p)**            | Non-compulsory      | - An action modification may refer to changing the association with an existing participant. For example OBD is no longer required to perform the action of BC sign off and another role will perform this action  
                    |                     |   - This may result in adding a new role to the process |
| **D(p)**            | Non-compulsory      | - As the association of process actions with participants are changed, it may not be required for the existing participant to remain in the process  
                    |                     |   - Thus it may result in deleting the previous roles from the process point of view |
| **M(p)**            | Non-compulsory      | - The above-mentioned scenarios of deletion and addition of participants because of action modification are extreme situations.  
                    |                     |   - However, most likely scenario would be the modification of responsibilities of existing participants |
| **A(d)**            | Non-compulsory      | - When actions are modified, it could mean the changing the action interfaces associated with a process.  
                    |                     |   - Change of action interfaces, may result in addition of new data elements to the business object of the process |
| **D(d)**            | Non-compulsory      | - Due to the changes in action interfaces, it may be decided to remove certain existing data elements from the business object of the process |
| **M(d)**            | Non-compulsory      | - Similar to both situations of addition and deletion of data, a change in the action interface may result in having to modify data elements of the business object as a consequence of modification of an action |
| **A(p)**            | Non-compulsory      | - For example adding a subcommittee at the school level to be responsible for the academic quality of courses, could be considered as an addition of a new participant  
                    | (Depicted Figure 7.4) |   - This may result in having to get the proposed CNI approved through this sub-committee, which leads to adding an action prior to the submission of CNI by Head of School |

Chapter 7-Nature of Process Element Evolution
<table>
<thead>
<tr>
<th>Initiating evolution</th>
<th>Propagating Change</th>
<th>Compulsory or non-compulsory</th>
<th>Description and/or Example</th>
</tr>
</thead>
</table>
| M(a)                 | Non-compulsory     | - If a sub-committee is appointed to ensure the academic quality of courses, it may be decided to give the school level approval to that committee instead of the Head of School.  
- This is modification of actions arising from addition of a participant |
| D(p)                 | Non-compulsory     | - When certain roles are added and responsibilities are assigned to the new roles, some of the existing roles may become obsolete.  
- This may result in removing the existing roles. |
| M(p)                 | Non-compulsory     | - Similar to above deletion, as new roles are added, existing roles may be modified by changing responsibilities or reporting structures associated with them. |
| D(p) (Depicted Figure 7.4) | D(a) | Non-compulsory | - When participant associated roles are deleted, specific actions associated with those participants may no longer required  
- This may give rise to a deletion of certain actions associated with the process. |
| M(a)                 | Compulsory         | - When roles are deleted, from an organisational structure, such as removing the role of OBD, it would require certain modifications to take place in the process to assign that task to some other participant. |
| M(p)                 | Non-compulsory     | - For strategic or operational reasons if the role of OBD is removed form the organisation, certain responsibilities associated with the deleted role may be assigned to other existing roles.  
- Thus a deletion of a participant may lead to modification of another participant. |
| M(p) (Depicted Figure 7.4) | A(a) | Non-compulsory | - Modification of participant may mean changes in responsibilities or reporting structure associated with that role.  
- This could give rise to situations of having to add new actions to the process, to cater for this situation. |
| D(a)                 | Non-compulsory     | - Consider the example of change of reporting structure of Head of School having to directly report to PVC-A, without going through the Dean.  
- This may result in deleting the action ‘Submit CNI’ action from the process, as Dean is no longer sponsoring the course. |
| M(a)                 | Non-compulsory     | - Consider a scenario where responsibility of assuring the academic quality is no longer required to be checked by the Dean.  
- In this case, it may result in having to modify the action interface that Dean interacts with, to change the visibility of business data, in order to give more prominence to information that s/he is required to validate. |
<p>| A(p)                 | Non-compulsory     | - Changes such as responsibility or reporting structures of participants may result in having to introduce new roles to take care of certain responsibilities. |</p>
<table>
<thead>
<tr>
<th>Initiating evolution</th>
<th>Propagating Change</th>
<th>Compulsory or non-compulsory</th>
<th>Description and/or Example</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>D(p)</td>
<td>Non-compulsory</td>
<td>- If a role of the Dean is modified to have the executive responsibility of having to assure the cost of courses, it may no longer required to have the services of OBD performing the action of ‘signing off the BC with independent assessment’</td>
</tr>
<tr>
<td></td>
<td>M(p)</td>
<td>Non-compulsory</td>
<td>- Since participants are associated to each other through reporting or delegation authorities, a change to one process role may result in having to modify another.</td>
</tr>
<tr>
<td>A(d) (Depicted Figure 7.5)</td>
<td>A(a)</td>
<td>Non-compulsory</td>
<td>- The addition of data could be considered from adding one data element to a report. - Consider the scenario of having to add market analysis data to support the BC for the course. - The verification of market analysis data is best done by a party who are experts in that. - As such a new action would be added to verify market analysis data in the BC approval process</td>
</tr>
<tr>
<td></td>
<td>M(a)</td>
<td>Compulsory</td>
<td>- When changes are introduced to the underline business object, it is inevitable for the action interfaces to be changed to reflect those changes. - Therefore from the data visibility point of view, actions are required to be changed</td>
</tr>
<tr>
<td></td>
<td>M(d)</td>
<td>Non-compulsory</td>
<td>- When new data items are added, the integrity or computational relationships of exiting data may change. - Thus as addition of one data element may result in modifying existing data elements</td>
</tr>
<tr>
<td></td>
<td>D(d)</td>
<td>Non-compulsory</td>
<td>- When new data elements are added, it may no longer require keeping some existing data elements. - For example addition of fields, ‘course start date’ and ‘course end date’ may result in removing the attribute ‘course duration’ because the course duration now can be computed using the added new fields.</td>
</tr>
<tr>
<td></td>
<td>A(d)</td>
<td>Non-compulsory</td>
<td>- Addition of one data element such as ‘course costs’ may indicate the need for more data elements to calculate the real costs. These new elements would be ‘staff costs’, ‘facilities costs’, ‘overhead costs’ and ‘other miscellaneous costs’ - This indicate the possibility of having to add new fields by introducing another field</td>
</tr>
<tr>
<td>D(d) (Depicted Figure 7.5)</td>
<td>D(a)</td>
<td>Non-compulsory</td>
<td>- Consider the example of deleting fee information associated with a BC. In this case if there were any particular actions in place to collect this data, it would no longer be required in the process - Thus a deletion of data components may lead to deletion of actions</td>
</tr>
<tr>
<td></td>
<td>M(a)</td>
<td>Compulsory</td>
<td>- Deletion of business object data inevitably needs to be reflected in action interfaces. - For example, deletion of course description from the business object, will result in removing those fields from the web forms that collect and displays that information</td>
</tr>
<tr>
<td></td>
<td>M(d)</td>
<td>Non-Compulsory</td>
<td>- Deletion of the fields that has integrity relationships or computational relationships, may result in having to change the existing data element to effectively reflect the change</td>
</tr>
<tr>
<td>Initiating evolution</td>
<td>Propagating Change</td>
<td>Compulsory or non-compulsory</td>
<td>Description and/or Example</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------</td>
<td>-----------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>D(d)</td>
<td>Non-Compulsory</td>
<td>- Deletion of cost Figures as it is no longer required from the BC related data, will result in deleting the total cost Figures as well. - This shows the possibility of deletion of one set of data leading into deletion of another.</td>
<td></td>
</tr>
<tr>
<td>A(d)</td>
<td>Non-Compulsory</td>
<td>- The deletion of 'course duration' field result in having to add two more fields as 'course start date' and 'course end date' - This shows the possibility of deletion of one element of data that could lead to addition of another.</td>
<td></td>
</tr>
<tr>
<td>M(a)</td>
<td>Compulsory</td>
<td>- Modification of data by changing its format or relationships has a direct correlation with action interfaces.</td>
<td></td>
</tr>
<tr>
<td>M(d)</td>
<td>Non-Compulsory</td>
<td>- Due to the computational or integrity relationships among data elements, changes from integer to float in total cost Figures, needs to be reflected in the other fields that helps calculating the total cost.</td>
<td></td>
</tr>
<tr>
<td>D(d)</td>
<td>Non-Compulsory</td>
<td>- Modification such as associating one unit with one course can lead to the deletion of certain fields maintained to manage the many to many relationship.</td>
<td></td>
</tr>
<tr>
<td>A(d)</td>
<td>Non-Compulsory</td>
<td>- Modifications to data elements such as computational relationships may result in having to add new data elements for completeness.</td>
<td></td>
</tr>
</tbody>
</table>

There were 43 different types of propagating impacts identified in Table 7.3 above. Out of these 43 propagating evolutions, only six are compulsory evolutions. These compulsory evolutions are interestingly all resulting in modification of actions. The suggested compulsory change are; addition of an action –A(a), deletion of an action – D(a), deletion of a participant – D(p), addition of data elements – A(d), deletion of data elements – D(d), and modification of data elements – M(d).

The propagating impact not only depends on the nature of the evolutions (primitives, semantics, and dynamics) but also the positioning of the changing element in the bigger process. Based on the positioning of the changing element, the other elements in the process have impacts of varying degree. This severity of impact is discussed under the topic DISN impact in the next section.

### 7.5 DISN Impacts of Process Element Changes

For both compulsory and non-compulsory changes, the impact on other elements not only depends on the nature of the change, but also the element associations with other elements and its positioning in the process. For example, if a
changing action that is in the very beginning of the process in a sequential arrangement, most of the subsequent actions might be affected. On the contrary, if a latter action is changed or if that action is in a conditional branch there are chances that most of the other actions being able to proceed without any issues.

Here a classification is provided to represent different kinds of propagating impacts. This classification recognises four categories named Direct, Indirect, Secondary, and Non-cautionary impacts (DISN impact) as follows;

- **Direct** impact refers to particular guard elements or actions in a process that is directly affected due some change.

- **Indirect** impact denotes particular guard elements or actions in a process, which cannot be computed or executed, due to the direct impact on some other elements.

- **Secondary** impact indicates particular guard elements or actions that can be computed or executed, but cannot be merged with the rest of the process for successful completion. This is also caused by direct impacts.

- **Non-cautionary** impact refers to the guard elements and actions that logically appear before the point of change or in a parallel branch, which has no or minimal impact.

For the identification of DISN impact, this research uses the linear process expressions created using Kleene Algebra with Test (KAT). Chapter-6 introduced a series of KAT based definitions to conceptualise all possible correlations among process elements. KAT based linear processes expressions created cohesively and completely capture all types of associations among process elements.

Figure 7.7 shows a hypothetical process example of a leave approval process.

![Figure 7.7- Hypothetical Leave Approval Process](image)
The actions associated with the leave approval process depicted in Figure 7.7, could be described as follows:

- $\alpha_1$ – The applicant fills in the leave approval form and forward to the immediate manager
- $\alpha_2$ – Immediate manager checks the leave approval form
- $\alpha_3$ – If the leave is for annual leave the quality officer approves the leave form
- $\alpha_4$ - If the leave is for sick leave the line manager approves it
- $\alpha_5$ – Alternate staffing is arranged by the line manager
- $\alpha_6$ – Alternate staffing is checked by the quality officer
- $\alpha_7$ – Departmental Manager checks the leave form and alternate staffing arrangements
- $\alpha_8$ – Shift planners re-schedule the shift work until it allows them to meet production targets
- $\alpha_9$ – Final approval for leave is given by the production manager

This ‘leave approval process’, has the above nine process actions ($\alpha_1$ to $\alpha_9$) and following guard elements:

- $\varphi_1 \rightarrow P$ (applicant, $\alpha_1$) – The applicant is permitted to perform action $\alpha_1$
- $\varphi_2 \rightarrow V$((DATA=>{defines the fields to be displayed in the leave form according to the Definition-12 in Chapter-6 }), $\alpha_1$)
- $\varphi_3 \rightarrow O$ (immediate manager, $\alpha_2$) – The immediate manager is obligated to perform action $\alpha_2$
- $\varphi_4 \rightarrow $ PC (applicant, immediate manager, (characteristic => { report to })))
- $\varphi_5 \rightarrow P$ (quality officer, $\alpha_3$)- Here $\varphi_5$ indicates that ‘quality officer’ is permitted to perform the action $\alpha_3$.
- $\varphi_6 \rightarrow P$ (manager, $\alpha_4$)- Here $\varphi_6$ denotes that ‘manager’ is permitted to perform the action $\alpha_4$
- $\varphi_7 \rightarrow O$ (quality officer, $\alpha_6$)- Here $\varphi_8$ denotes that ‘quality officer’ is obligated to perform the action $\alpha_6$.
- $\varphi_8 \rightarrow O$ (departmental manager, $\alpha_7$) – Departmental manager is obligated to perform action $\alpha_7$
- $\varphi_9 \rightarrow O$ (shift planners, $\alpha_8$) – Shift planners are obligated to perform action $\alpha_8$
- $\varphi_{10} \rightarrow O$ (production manager, $\alpha_9$) – Production manager is obligated to perform action $\alpha_9$
• \( \varphi_m \) → This is an internal condition used to manage the parallel processing
• \( \varphi_n \) → This is an internal condition used to manage the iteration of shift planning activity that checks production can meet targets

The KAT Expression 7.1 captures the process given in Figure 7.7 using the above described alphas and phis

\[
E = (\varphi_1 \varphi_2 \alpha_1) (\varphi_3 \varphi_4 \alpha_2) \varphi_m ((\varphi_5 \alpha_3 + \varphi_6 \alpha_4) + (\varphi_7 \alpha_5) \varphi_n (\varphi_8 \alpha_6))^* (\varphi_9 \alpha_7) \varphi_n (\varphi_{10} \alpha_8)^* (\varphi_{11} \alpha_9)
\]

**KAT Expression 7.1- KAT Expression for the process in Figure 7.7**

In relation to the above-described process, the following two hypothetical evolution scenarios are considered when discussing the DISN impact.

• **Evolution scenario 1**: The role ‘quality officer’ is removed from the organisational structure, due to a strategic reason.
• **Evolution scenario 2**: The role ‘manager’ is removed from the organisational structure, due to some strategic reason.

DISN impact for **Evolution scenario 1** is illustrated in Figure 7.8.

Here the removal of role of the participant is directly affecting the guard elements \( \varphi_5 \) and \( \varphi_8 \). This means that in the process execution, it is not possible to evaluate these guard elements, due to the non-existence of the role ‘quality officer’ in the organisational structure. This is shown using label **D- Direct** in Figure 7.8.
When the both guard elements $\phi_5$ and $\phi_8$ cannot be executed, the associated actions of those guard elements $\alpha_3$ and $\alpha_6$ have indirect impact as these cannot be reached because of directly impacted elements. This is denoted using label $I$: Indirect in Figure 7.8.

In Figure 7.8, label $S$: Secondary shows the action $\alpha_4$ that actually can be executed, but is not able to merge with the rest, for a successful completion of the process. This secondary impact takes place, because action $\alpha_4$ is in parallel with the direct impacted element, which cannot be executed.

The actions and guard elements that are logically prior to the directly impacted elements are denoted using, label $N$: Non-Cautionary. These elements are not affected due to the change given in evolution scenario 1.

![Figure 7.9- DISN impact of the Evolution Scenario 2](image)

Similar to the previous case the affected elements are labelled. According to Figure 7.9 there are only three - direct, indirect and non-cautionary impacts. There are no secondary impacts present. This is due to the positioning of changing element in the rest of the process, where the indirectly impacted element (action- $\alpha_4$) is in a conditional branch that permits other actions $\alpha_3$ to take place.

In this section, a categorisation of impacts is introduced, based on the positioning of the element being changed. The identified types are named as direct, indirect, secondary, and non-cautionary (DISN) impacts.

This notation of evolutions impact analysis and impacts categorisation in relation to business processes is not discussed in detail in previous work.
research that acknowledges the need for impact analysis is presented by Soffer (2005), Bodhuin et al. (2004), and Min, Bae, Cho, and Nam (1999). Among these works only Soffer (2005) offers a categorisation of impacts into two types as global or local. However, in the author’s opinion, the impact analysis is an important aspect in relation to reflecting process changes in web-workflow systems.

### 7.6 Chapter Summary

This chapter focused on comprehending the nature of business process evolutions, under *primitives, semantics, and dynamics of process element evolution*.

**Primitives of process evolutions** refer to *additions, deletions, and modifications* of process elements. When these primitives are applied to three process elements- *actions, participants* and *business object*, a combinatorial set of nine items were created.

**Semantics of process evolution** referred to comprehending the behaviour of process elements, in the presence of *evolution primitives*. As such, semantics of process evolution are studied in relation to: addition of actions-\(A(a)\), deletion of actions-\(D(a)\), modification of actions-\(M(a)\), addition of a participant-\(A(p)\), deletion of a participant-\(D(p)\), modification of a participant-\(M(p)\), addition of a business data object field-\(A(d)\), deletion of a business data object field-\(D(d)\), and modification of a business object data field-\(M(d)\).

**Dynamics of process evolution** discuss the nature of propagating impact of element changes. Findings reveal that there are 43 propagating changes resulting because of the above-mentioned nine types of evolutions. Out of those 43 propagating changes, only six are compulsory changes, requiring modifications to existing actions.

The propagating impact on other elements are categorised to identify the severity of changes required. On this basis, four categories are identified as *direct, indirect, secondary, and non-cautionary (DISN) impacts*. Impacts resulted based on the positioning of the changing element in the overall process. These *DISN impact* are located using KAT expressions that capture the positioning of process elements into linear process expressions.

Identification of *DISN impact*, in relation to an element change is presented in this chapter. This method involved manual analysis of the KAT expression of a
process. However, when processes are large and complex, it is impractical to do this analysis manually. Therefore, it requires an algorithm that can analyse KAT expressions to locate DISN impact of process element changes.

The next chapter presents an algorithm that can analyse KAT expressions to locate DISN impact. In addition, the major contribution of the next chapter is to develop the core solution to the problem of, effectively and efficiently reflecting process changes in web-workflow systems. This core solution is named as the PECIA (Process Evolution and Change Impact Analysis) Model. This model brings together, concepts and models, discussed and developed in Chapters-5, 6 and this chapter.
Chapter-8

“Intelligence is not information alone but also judgment, the manner in which information is collected and used”
Dr. Carl Sagan (1934-1996)

8 Process Evolution and Change Impact Analysis (PECIA) Model

8.1 Chapter overview

This chapter presents the Process Evolution and Change Impact Analysis (PECIA) Model. This model enables tracing impact of high-level process changes down to the implementation level artefacts. The PECIA Model integrates the schemata, data structures, and concepts (discussed in Chapter-5, 6 and 7) into a single framework.

The PECIA Model is structured around the framework named paradigm of process automation- PoPA framework; to provide a holistic approach to the problem of managing process evolutions in web workflow systems. The PoPA framework was initially discussed in detail in Chapter-1 (Figure 1.1 in page 5), to consist of four levels to represent processes as – pragmatic, semantic, syntactic and implementation level.

In relation to the above-mentioned four levels, two terms – intra and inter-level constraints, associations and dependencies (CAD) were discussed in Chapter-5. A number of data structures were developed to capture the intra and inter-level CAD, except for intra-level CAD in the semantic level.

In order to address the complexity of correlations among process elements, Chapter-6 was fully devoted to developing a formalism to capture intra-level CAD in the semantic level. The selected formalism was a bi-partite algebra named Kleene Algebra with Tests (KAT). A number of KAT definitions were created to capture CAD among process elements in Chapter-6. Using these definitions, correlations
among process elements are captured into cohesive linear expressions, suitable for analytical purposes.

Based on linear KAT expressions of a process, and the type of change that takes place in an element, Chapter-7 discusses the issue of locating the impact on other process elements. There four types of impacts were identified as direct, indirect, secondary and non-cautionary (DISN). In that discussion, a manual method was illustrated for locating DISN impacts of a process element changes and highlighted the need for an algorithm that can search the KAT expressions.

The first objective of this chapter is to develop an algorithm that can search the KAT expressions, represented in a binary tree, to locate DISN impacts. The second objective is to collate the data structures that capture both intra and inter-level CAD into a single model.

This chapter is organised as follows. Initially, the notion of change initiation and propagation is discussed in relation to the four levels of the PoPA framework. Following that, the PECIA Model is presented and its component parts are discussed in detail, to exemplify the use of each of them in locating propagating impact. Next, an algorithm is developed to search the tree structures that capture the KAT expressions to locate the DISN impacts. Finally, the usage of both the PECIA Model and the algorithm is exemplified using two evolution scenarios in relation to the New Course Approval Process automated in OCAS, which was introduced as the case study in Chapter-4.

### 8.2 Evolution Initiation and Propagation

Changes that affect a business process can get initiated at any level of the PoPA framework. Van der Aalst and Jablonski (2000) provide four possible reasons for changes of business processes as; i) changes to legal context, ii) changes in business or operational context, iii) logical and design errors, and iv) technical problems. These reasons accurately correspond to the four levels in the PoPA framework as follows:

- **Contextual information** artefacts in the pragmatic level change due to legal or strategic reasons.
• **Business process elements** in the *semantic level* change due to operational reasons.

• **Models** in the *syntactic level* are changed to solve logical or design errors.

• **Web-based workflow artefacts** in the *implementation level* may change due to technical problems.

Figure 8.1 shows the change initiation points and the propagation of changes across these levels. As indicated in the figure, the accumulated propagating changes at one level have a flow-on effect on the elements at the next level, due to inter-level correlations between elements at different levels. The sum of all accumulated changes need to be finally represented in the web-workflow artefacts at the *implementation level*.

![Figure 8.1 - Initiation of changes and Propagating Impact from Higher levels down to the implementation Level Artefacts](image)

In Figure 8.1, change initiation points are labelled from (A) to (D). The label (A) represents *contextual information* changes, such as a change in an organisational policy. Due to *intra-level CAD* at the *pragmatic level*, these changes first create impact on other contextual artefacts at the same level. Also, these changes in the
pragmatic level have a flow-on effect to process elements in the semantic level. In addition, there could be changes that get initiated at the semantic level, due to operational reasons (labelled as (B) in Figure 8.1). The complex correlations among process elements create DISN impacts on other elements at the semantic level. The aggregate all these changes at semantic and pragmatic levels, cause models to be changed at syntactic level. Further, at the syntactic level, changes to models are initiated due to design or logical errors (labelled as (C) in Figure 8.1). Finally, the total propagating impact that get initiated at implementation level (labelled as (D) in Figure 8.1) force amendments to the web workflow artefacts.

Comprehension of this propagating impact, based on intra and inter-level CAD, facilitates effective management of business process changes in web workflow artefacts. Therefore, the PECIA Model that facilitates managing both, intra and inter-level CAD, is presented in the next section.

8.3 Process Evolution and Change Impact Analysis (PECIA) Model

The PECIA Model (see Figure 8.2) brings together data structures that were previously developed in Chapters-5 and 6, capturing intra and inter-level CAD. This composite data structure consists of both relational schemata and binary-tree structure that represents linear KAT Expressions of a process. In this Figure 8.2, only entity level association is shown for the schemata.

There are four new entities introduced into the model, to keep track of evolutions that take place in elements. These entities are named as HistoryOfContextChange, HistoryOfElementChange, HistoryOfModelChange, and HistoryOfArtefactChange. These entities are shown as fully attributed entities, with appropriate relationships.

The PECIA Model also captures the nature of changes, based on the findings of Chapter-7. This nature of process element changes are captured into another fully attributed new entity named TaxonomyOfChange.
Figure 8.2- Process Evolution and Change Impact Analysis (PECIA) Model
As illustrated in Figure 8.2, there are three parts to the PECIA Model:

- Element Registries (A)
- Elements Mappings (B)
- Evolution Mappings (C)

The **Element Registries** (denoted as A) integrate all the *intra-level CAD* in the pragmatic, semantic, syntactic and implementation levels, into sub-layers labelled as A.1, A.2, A.3 and A.4 respectively. In particular, sub-layer A.2, which captures *semantic level CAD among process elements*; encapsulates KAT expressions presented in binary-tree structures.

The **Element Mappings** section (referred as B) captures three inter-level associations between- pragmatic and semantic, semantic and syntactic, and syntactic and implementation levels into three sub-layers denoted as B.1, B.2 and B.3 respectively.

The **Evolution Mappings** section facilitates in recording process changes. This recorded information supports the formation of queries for locating propagating impact. In addition, this section also houses the referential entity (TaxonomyOfChange), which records nature of process element evolutions introduced in Chapter-7.

### 8.3.1 Methodology for using the PECIA Model

The methodology shown in Figure 8.3, shows the steps involved using the PECIA Model. This methodology is adapted based on Castano et al.’s (1999) work on process analysis and re-engineering.

As shown in Figure 8.3, prior to the initial use the model, it needs to be populated to capture meta-information related to the process automation. In other words, both *inter* and *intra-level CAD* of a particular process. Once the model is populated, it can be used to find the impact of process evolutions. In finding the propagating impact, evolution scenarios are iteratively run through the PECIA Model, until the full set of propagating impacts are identified. Once the full set of changes is identified, those changes then are appropriately reflected in web-workflow artefacts or models.
At present, the PECIA Model needs to be populated by a human who comprehends the automation of a particular process. In other words, there is yet no automated mechanism to capture the intra and inter-level CAD into the PECIA Model. For the discussion of this chapter, it is assumed that the model is accurately populated to reflect all possible intra and inter-level CAD.

### 8.4 Using the PECIA Model to Locate Propagating Impact

This section discusses the use of the PECIA Model based on four evolution initiation points at, pragmatic, semantic, syntactic, and implementation levels. The use of each entity in the PECIA Model (Figure 8.2) will be exemplified in this discussion.
8.4.1 Handling Changes at the Pragmatic level

The high-level UML activity diagram in Figure 8.4 shows the steps involved in managing contextual information element changes at the pragmatic level. In this diagram, there are three swim-lanes to denote, the **PECIA Model**, a team of humans made up of business analysts and process managers who are responsible for making decisions, and other systems where contextual artefacts are maintained.

![Figure 8.4- Steps in handling Changes at the Pragmatic level](image)

The states A, B, C and D correspond to places in which changes can initiate. As illustrated in Figure 8.4 there are five steps involved in finding propagating impact. These steps are discussed in the logical order as they appear in Figure 8.4.

8.4.1.1 Step 1 - Changes to Contextual Artefacts

Both external and internal contextual information artefacts change due to strategic and legal reasons. Usually business analysts, process managers or other relevant administrative staff are informed of such changes via organisationally established communication methods such as memorandums, newsletters, notification
e-mails or announcements. Then these changes are expected to be reflected in appropriate systems, processes, or documents. This step denotes such changes taking place in contextual artefacts, outside the web-based workflow system.

8.4.1.2 Step 2 - Identify Changing Contextual Artefacts

Contextual artefact change described in step 1 is external to the operational processes. Therefore, upon the notification of a contextual artefact change, described in step 1, the applicability of that change to the process needs to be ratified by the process managers, which is denoted in this step.

8.4.1.3 Step 3 - Temporary Record of Contextual Artefact Changes

Once the changing contextual artefact is identified a record is made in the entity named \textit{HistoryOfContextChange} in the \textbf{PECIA Model} (see Figure 8.2). The underlying SQL statement used would be similar to following.

\begin{verbatim}
UPDATE HistoryOfContextChange SET <<--full attribute list of the entity and the appropriate values-->>;
\end{verbatim}

8.4.1.4 Step 4 - Find propagating impact on other Contextual Artefacts

Once a temporary record is created (in previous step 3), a search is carried out to find \textit{intra-level} impact propagation. A query similar to following would be run for this purpose.

\begin{verbatim}
SELECT <<--both fragment IDs and contextual artefact IDs-->> FROM fragmentContextAssociations WHERE <<--ID is similar to already changed contextual artefact’s ID in previous step-->>;
\end{verbatim}

In this query, associations are identified by searching the pre-populated entity \textit{fragmentContextAssociations} present in layer A.1 of the \textbf{PECIA Model} (Figure 8.2).

8.4.1.5 Step 5 - Prompting other Impacted Contextual Artefacts

Next step is to prompt the list of all the associated contextual artefacts and fragments that are impacted due to the original contextual artefact change. This list would be prompted to an output screen similar to Figure 8.5. This would allow process managers to decide the artefacts that require further changes.
As shown in Figure 8.5, if elements are selected for further changes, the process, in the activity diagram in Figure 8.4, will repeat from steps-1 to 5. If no further changes are required, that is if the ‘Continue No Changes Required’ button (in the output screen in Figure 8.5) is pressed, the process moves to the next stage of locating intra-level CAD at semantic level.

8.4.2 Handling Changes at the Semantic level

Elements at semantic level can change due to flow-on effects from contextual element changes and/or for operational reasons in the real-life business process. Figure 8.6 illustrates the steps involved in handling both types of changes.

As indicated in Figure 8.4, all four states are shown in boxes labelled from A to D. In Figure 8.6 expands the procedural steps carried out in state B, which denote changes at semantic level.
### Chapter 8 - Process Evolution and Change Impact Analysis (PECIA) Model

#### Figure 8.6 - Steps in handling Changes at the Semantic level

<table>
<thead>
<tr>
<th>PECIA Model</th>
<th>Humans who carry out the changes</th>
<th>Workflow System / Other External System</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(A) Pragmatic Level Artefact Changes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(B) Semantic Level Element Changes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(6) Temporary Contextual Information Change Entry</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(10) Find Propagating impact of Process elements change onto other elements</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(11) Prompt the list of affected other process elements</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(C) Syntactic Level Model Changes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(D) Implementation Level Workflow Changes</td>
<td></td>
</tr>
</tbody>
</table>

There are six actions identified and these are numbered as steps-6 to 11; with numbers continuing from the previous Figure 8.4. Each of these steps is discussed next, according to the order of numbering in Figure 8.6.

**8.4.2.1 Step 6 - Finding Process Elements affected by Context Changes**

This step would be invoked only if changes are flowing because of contextual element changes. In this case, the aim is to locate a list of impacted process elements and prompt a list for business analyst to decide on the necessity of process element changes. A query similar to the following could be executed for this purpose.
The above query uses the entity ContextProcessAssociation, given in the element mapping layer named as B.1 (the PECIA Model in Figure 8.2), to identify possible associations between process elements and contextual artefacts. Then based on the association recorded in ContextProcessAssociation, impacted element details are extracted from ProcessElementRegistry entity, given in layer A.1 in Figure 8.2.

The extracted data is presented to the business analyst in an output similar to Figure 8.7, to identify further changes required to process elements.

![Figure 8.7- List of process elements affected due to Contextual Artefact changes](image)

Not every contextual artefact change has impact on process elements. However, given the associations (Figure 8.7) the business analyst can make an informed decision. If the business analyst finds that changes are not required to process elements, the process will end without proceeding to next step. If the business analyst decides that changes are required to associated elements, the process moves to the next step.
8.4.2.2 Step 7 - Changes to Operational Processes

Changes to operational processes can get initiated in two ways: a) as a flow on effect from previous step-6, and/or b) due to operational reasons that take place in the real-life processes (van der Aalst et al., 2000). Whichever way, the participants of the process are required to be informed about the new changes to the process. This would generally occur via process managers who sent out a memorandum informing the changes.

Changes that take place in operational processes have an impact on the implemented systems or some other processes. This leads to business analysts needing to identify the changes required in workflow systems in the next step.

8.4.2.3 Step 8 - Identify Changing Process Elements

This step refers to business analysts identifying the automated elements of a process that may require changes. When an element is identified, it also requires classifying the type of change that is taking place. The primitives of changes that can take place in process elements were identified to be additions, deletions, and modifications, previously in Chapter-7. In addition based on the element that is evolving, the semantics of the change can be determined by the business analyst. Semantics of process element evolutions were also discussed in Chapter-7.

8.4.2.4 Step 9 - Temporary Record of Process Element Change

Once business analysts have decided on the changes required to particular process elements, this change is temporarily recorded in the PECIA Model. For this purpose, a SQL statement similar to the following will be activated.

```
UPDATE <<-- entity HistoryOfElementChange -- >> SET <<--
appropriate values for all attributes in the entity -->>;
```

This SQL statement makes a temporary entry in HistoryOfElementChange entity, shown in evolution mapping layer (labelled as C) in the PECIA Model (Figure 8.2). This temporary record will be used to locate propagating impact in the next step.
8.4.2.5 Step 10 - Finding the Propagating Impact of the Element Change

This step deals with finding propagating impact of process element change on other elements. For this purpose, linear KAT expressions of processes are used. Apart from KAT expressions, primitives and semantics of elements evolutions, which are recorded in the previous step-9 are also taken into consideration. This searching requires an algorithm that can analyse binary-trees that represent KAT expressions.

Due to the relative length of the discussion, an algorithm that can locate DISN impacts is separately discussed in section 8.5 of this chapter. The expected outcome of this algorithm is a list of process elements that falls under each category (direct, indirect, secondary, and non-cautionary) of DISN impacts.

8.4.2.6 Step 11 - List of Impacted Process Elements

This is the final step in handling propagating impact at semantic level. This step lists impacted elements as direct, indirect, secondary, or non-cautionary impacted elements. Business analysts would need to focus on rectifying direct impacted elements. This would move the process back to step-7 of the activity diagram given in Figure 8.6. If there are no directly impacted elements, the process will proceed to the next level of propagation impact analysis in the syntactic level.

8.4.3 Handling Changes at the Syntactic Level

At the syntactic level of the PoPA framework, changes can get initiated due to logical or design errors (van der Aalst et al., 2000). In addition, process element changes create a propagating effect on models at the syntactic level. Figure 8.9 illustrates steps involved in handling changes at the syntactic level.

There are six steps, in handling model changes at the syntactic level (Figure 8.9); that are similar to the steps at the semantic level (Figure 8.6). The step numbering (in Figure 8.9) continues from previous Figure 8.6. Each of these steps is discussed next, according to their ordering.
8.4.3.1 Step 12 - Finding Impacted Models due to Element Changes

The last step at *semantic level* (step-11 in previous stage) would pass a list of process elements that were changed into this step. Based on this list of changed process elements, the *PECIA Model* is searched to locate the models that are associated with changed process elements. A query similar to the following would be used for this purpose.

```
SELECT <<-- Model Details -->> FROM << from the entity named ModelRegistry -->> WHERE <<-- condition that checks whether the elementOrCId in ProcessModelAssociation entity is similar to the one that was changed in the previous step -->>;
```

Details of the models in the *syntactic level* are extracted from *ModelRegistry* entity depicted in layer A.2 in Figure 8.2. This search uses the process element and model associations recorded in the entity named *ProcessModelAssociation* for this purpose.

Extracted details of affected models are prompted to the business analyst in an output screen similar to the one in Figure 8.9.
Here a decision is made to carry out further changes to the models, which are associated with the changed process elements. If models require changes, the process would move to step-13 or otherwise terminate, as shown in Figure 8.8.

8.4.3.2 Step 13 - Changing the Models

Changes to models do not always flow from the previous step. In some situations, changes can be initiated at models, to rectify previous design or logical errors.

In process automation, there are a number of models created to capture data, processes and interface definitions. Out of these models, generally machine-readable process models are used to enact the workflow at runtime. Therefore, carrying out changes accurately in these models are vital. Models can be maintained via editing tools such as Rational Rose, Eclipse, and Visio, that can support visual models. Once changes are carried out in these models, they are appropriately used to either, auto-generate changed web-workflow artefacts, or manually reflect changes.

8.4.3.3 Step 14 - Identify Changing Models

After models have changed, business analysts require identifying whether any other models are impacted on due to this particular change. For this reason, first he/she needs to identify the changed models and record them in the PECIA Model.
8.4.3.4 Step 15 - Temporary Record of Model Changes

Once changing models are identified, a temporary record is be made in the **PECIA Model** to reflect this change. For that, a SQL statement similar to the following is be executed.

```
UPDATE << -- entity HistoryOfModelChange--->> SET <<--
attributes and appropriate values to reflect the changes to the models -->>>;
```

A temporary record is made in the entity *HistoryOfModelChange*, present in *evolution mapping layer* of the **PECIA Model** (Figure 8.2).

8.4.3.5 Step 16 - Finding Propagating Impact of the Model Changes

Models are associated with each other and their associations are recorded in the **PECIA Model**. Here, based on that recorded association, following query is used to find other models that are impacted.

```
SELECT <<-- affected other model details -->> FROM <<--
ModelRegistry-->> WHERE <<--condition that checks whether the modelID in ModelAssociations is similar to the one that was changed in the previous step -->>;
```

Here the entity *ModelAssociations* present in layer labelled A.3 of the **PECIA Model** (Figure 8.2) is searched to finding propagating impact of model changes recorded in the previous step. Impacted model details are derived from the entity named *ModelRegistry*.

8.4.3.6 Step 17 - List of Impacted Process Elements

An output interface similar to Figure 8.10 prompts with the list of models affected. Here if business analysts decide to change the affected model, process iterate from step 14 in Figure 8.8. Otherwise, the changes will continue to the next stage.
8.4.4 Handling Changes at the Implementation level

This is the last stage in the process of finding propagating impact of changes. In this stage, the aim is to reflect all the propagated changes in web-based system artefacts.

Similar to previous stages, changes to workflow artefacts can happen due to two reasons. First, changes could be the propagating from upper level changes or changes could be initiated at this level to fix technical errors of previous implementation (van der Aalst et al., 2000). Figure 8.11, shows steps in handling both these changes at the implementation level of the processes.
When handling changes at *implementation level*, there are ten steps identified. Out of these ten tasks, the first six steps are similar to the steps in previous stages. The last four tasks are specific to handling changes in implemented systems. Each of these steps is discussed below.

### 8.4.4.1 Step 18 - Impacted Workflow Artefacts due to Model Changes

This step is actioned only if the changes are flowing from the upper levels. Here, particular attention would be given to finding impacted workflow artefacts, due to the changes introduced to models at the *syntactic level*. A query similar to below could be used for this purpose.

---

**Figure 8.11 - Steps in handling changes at the Implementation Level**

When handling changes at *implementation level*, there are ten steps identified. Out of these ten tasks, the first six steps are similar to the steps in previous stages. The last four tasks are specific to handling changes in implemented systems. Each of these steps is discussed below.

### 8.4.4.1 Step 18 - Impacted Workflow Artefacts due to Model Changes

This step is actioned only if the changes are flowing from the upper levels. Here, particular attention would be given to finding impacted workflow artefacts, due to the changes introduced to models at the *syntactic level*. A query similar to below could be used for this purpose.
SELECT <<-- web artefact details -->>, FROM <<-- entity WebArtefactRegistry -->>, WHERE <<-- condition that checks whether the ModelID in ModelWebArtefactAssociation entity is similar to the one that was changed in the previous step -->>;

Entity named WebArtefactRegistry (A.4 layer in the PECIA Model), which keeps track of all models used in the workflow implementation. The association between workflow artefacts (recorded in entity WebArtefactRegistry) and models in the syntactic level is captured into an entity named ModelWebArtefactAssociation in layer B.3 of the PECIA Model (Figure 8.2). Based on these recorded associations, a list of web artefacts is extracted using the above query.

Information extracted by running the above query could be presented in an output similar to Figure 8.12.

![Figure 8.12- Showing the Web Artefacts that are affected by a model change](image)

If the business analyst decides that these suggested workflow artefacts (Figure 8.12) require changes, the process would proceed to next step. If none of the elements are selected for further changes, the process terminates at this point.

### 8.4.4.2 Step 19 - Changing Web-based Workflow Artefacts

Once the list of affected artefacts is provided (Figure 8.12) system administrators or programmers, could carry out changes as required. In addition,
changes could be initiated at this point, due to technical errors in the implemented systems. Reflecting both these changes in workflow systems is denoted in this step.

8.4.4.3 Step 20 - Identify Changing Workflow Artefacts

Once a change is made into a web-workflow artefact, it requires a guarantee that this change has not caused further implications to the system. Therefore, this change has to be recorded in the **PECIA Model**. Identification of changes to be recorded in the **PECIA Model** is indicated in this step.

8.4.4.4 Step 21 - Temporary Record of Web Artefact Changes

Once changing workflow artefacts are identified, SQL statement similar to the following is used to make a temporary record in the **PECIA Model**.

```
UPDATE <<-- entity HistroyOfWebArtefactChange--->> SET <<--attributes and appropriate values to reflect web workflow artefact change-->>;
```

As shown in the query above, a record would be made to the entity named *HistroyOfWebArtefactChange* in the evolution-mapping layer of the **PECIA Model**.

8.4.4.5 Step 22 - Propagating Impact of Workflow Artefact Changes

Workflow artefacts are associated with other artefacts; therefore changes could create propagating impact on these associated artefacts. The artefacts that are probably affected can be found using a query similar to below.

```
SELECT <<-- artefact details-->> FROM <<-- entity WebArtefactRegistry -->> WHERE <<-- condition that checks the association between workflow artefacts based on entity named artefactAssociation -->>;
```

This query extracts other associated and impacted workflow artefact information, from the entity named *WebArtefactRegistry*, in the layer B.4 of the **PECIA Model**, conditioned by the associations recorded in *artefactAssociation* entity.
8.4.4.6 Step 23 - Prompt the List of Artefacts Possible to be Affected

The output of this step lists the implementation artefacts that are possibly affected in an interface similar to Figure 8.13. By closely analysing the given list, business analysts can decide whether these artefacts need further changes.

![Image: List of Web workflow Artefacts Possible to be affected](image)

*Figure 8.13- List of Web workflow Artefacts Possible to be affected*

If further changes are required to other web workflow artefacts, the process repeats from step-19, or otherwise the process moves to the next step.

Previous steps discussed (step-1 to step-23), facilitated carrying out changes that could initiate at any level of the PoPA framework, down to the implementation artefacts of the web workflow system. In these steps, business analysts could make informed decisions to change (or not to) elements in each level as all correlated elements are systematically extracted and prompted. These associated elements are found by querying correct entities (in the PECIA Model-Figure 8.2) that record both *intra* and *inter-level* correlations of elements at pragmatic, semantic, syntactic, and implementation levels.

After changes are reflected in workflow artefacts, it still requires a few other steps to be performed, for a successful completion of the process. Steps-24 to 27 detailed below, carryout these completion tasks such as making temporary records permanent and managing process instances of which definitions are changed.
8.4.4.7 Step 24 - Testing the Changes (Evolution Analysis)

Once changes have taken place in processes, it is vital to test the changes for their accuracy from the technology point of view. This aspect is referred to as workflow evolution analysis (Weske, 2000). In workflow evolution analysis three types of analytical aspects – verifications, validations and performance analysis are encouraged to be carried out (van der Aalst, 2002). These workflow analytical aspects are specific to a particular workflow modelling language or implementation.

The evolution analysis under verification, validation, and performance analysis is considered to be out of scope of this research. However, workflow analysis is acknowledged as vital for successful completion of the evolution process. Hence, this step indicates carrying out appropriate workflow evolution analysis in order to assure syntactical correctness or performance of the process.

At the completion of this step, if any errors are found, the process of evolution is required to move back to the appropriate evolutions stages A, B, C or D as indicated in Figure 8.11. If errors are not found the evolution process moves to the next step.

8.4.4.8 Step 25 - Process Instance Transfers

When process definitions change, the running instances of that process needs to be handled in an appropriate manner, referred to as dynamic changes in workflow related research (Casati et al., 1998). There are several methods acknowledged to be used for handling dynamic changes, such as: forward recovery, backward recovery, proceed, transfer and detour (van der Aalst et al., 2000).

This step indicates making dynamic changes, which involves identifying and transferring the process instances that are affected due to workflow changes. However, this aspect of dynamic changes is considered to be out of scope of this research. Therefore, it is not intended to discuss methodologies and techniques in relation to dynamic changes and assumes that such changes are carried out successfully.

8.4.4.9 Step 26 - Identify all the Temporary Changes made

Thus far in the evolution process (from step-1 to step-24), all the changes that took place in external systems to the PECIA Model are recorded as temporary
entries in the model. At this point, the business analyst would identify all temporary changes.

8.4.4.10 Step 27 - Commit Temporary Changes to the PECIA Model

As the name of this step indicates this step would commit all temporary records, to be permanent records in the following entities in evolution mapping layer in the PECIA Model (Figure 8.2).

- **HistoryOfContextChange** - that records changes to contextual artefacts in the pragmatic level
- **HistoryOfElementChange** – that records changes to process elements in the semantic level
- **HistoryOfModelChange** – that records changes made to all types of models
- **HistoryOfWebArtefactChange** – that records changes to all web-workflow artefacts

These change records could be used as future evolution analysis purposes such as discovering patterns of changes associated with process elements.

This section discussed the steps related to carrying out changes that could initiate at four levels (pragmatic, semantic, syntactic and implementation) down to workflow artefacts. This evolution process is facilitated by the PECIA Model (Figure 8.2), which captures both intra and inter-level correlations among elements in aforementioned four levels.

In relation to this evolution process, 27 steps were discussed in detail. However, in this discussion, one particular step (step-10) was postponed to be discussed later, due to its relative complexity. This step is related to handling evolutions in the semantic level; in particular, to find the propagating impact of process element changes using KAT based process expressions. Therefore, the next section details an algorithm created for identifying DISN (direct, indirect, secondary, and non-cautionary) impacts using KAT based process expressions.
8.5 An Algorithm to Find Propagating Impact of Process Element Changes

Chapter-6 of this thesis detailed a mechanism to capture correlations among process elements into KAT based linear process expressions. These KAT expressions mainly consist of two types of elements- *alphas* and *phis*. *Alphas* represent process actions and *phis* captured all other associations among process elements. Linear KAT expressions were represented as binary-trees, which facilitate easy manipulation of elements for analytical purposes.

This section details an algorithm to locate the propagating impact of an element change onto other elements, using binary-tree representation of KAT expressions. The objective of this algorithm is to locate the following four types of impacts on other process elements:

- **Direct impact** – Phis or alphas in KAT expression that are directly affected because of a suggested change.

- **Indirect impact** – Phis or Alphas that do not have any logical errors but are not able to be researched due to directly impacted elements.

- **Secondary impact** – Phis and Alphas that are reachable and executable, but unable to merge with the main flow for successful completion of the process.

- **Non-cautionary**– These are process elements that have minimal or no impact due to changes that have taken place.

The above *DISN impacts* take place in two phases. First *directly impacted* elements are located. Then based on the position of the directly impacted element in the KAT expression, affected *phis* and *alphas* are located.

To locate *directly impacted phis* and *alphas*, *ProcesElementRegistry* entity in layer A.2 of the **PECIA Model** (Figure 8.2) is searched. A query similar to following would be used for locating directly impacted process elements.
SELECT <<--phi or alpha notations-->> FROM <<--
ProcessElementRegistry-->> WHERE <<--condition that
checks the association of changing process elements
artefacts (i.e. action name, participant identified in
roles, or object attribute names), associated with the
selected phi or alpha -->>;

Once directly impacted \textit{phi} or \textit{alpha} is identified then the binary tree
representation of KAT expression is used to locate the \textit{indirect}, \textit{secondary}, and \textit{non-
cautionary} impacts.

A binary tree representation of a process is visualised in Figure 8.14 below.
Figure 8.14 represents a random process and details of \textit{phis} and \textit{alphas} are
unimportant at this point.

![Figure 8.14- Binary Tree Representation of KAT expression of a particular Process](image)

A tree structure similar to above is created based on a number of invariants,
which were previously explained in Chapter-6. A binary-tree representation of a
process can have a non-determinable number of levels, depending on the complexity
of the process. The directly impacted \textit{phi} or \textit{alpha} identified from the previous query
could be anywhere in this structure. Therefore, searching for \textit{indirect}, \textit{secondary}, and
\textit{non-cautionary} impacted elements has to be carried out in a systematic manner,
using an appropriate algorithm, which can be applied with any process
representation.

The following are the algorithmic-steps (at the highest level) suggested for
searching a binary tree structure similar to Figure 8.14.

- **Algorithmic-Step-1**: Locate the position of directly impacted process
element by traversing the binary-tree using in-order traversal method
(Shave, 1975; Knuth, 1973), which allows reading all elements only once.

- **Algorithmic-Step-2**: When the changing element is located (its position in the binary tree), then traverse upward until the root is reached.

- **Algorithmic-Step-3**: In every step in this upwards traversal, *phis* and *alphas* in left and right branches of the tree are marked either as in indirect, secondary, or non-cautionary impacted elements.

In *Algorithmic-Step-3* above, a number of factors are taken into consideration when deciding the type of impact. These factors are:

- Whether the directly impacted element is a *phi* or an *alpha*
- Whether this changing element is on left or right side branch of the tree
- The type of the notation (phi, alpha or the type of operator plus (+), dot (.), star (*) or negation (~)) held in the parent node
- The type (phi or alpha) of sibling node and its children nodes

This complex logic involved in identifying the impacts on other elements is summarised in Table 8.1. There are mainly three sections in this table. The first section refers to *Algorithmic-Step-2* of the high-level algorithm, which checks the type of parent nodes and sibling nodes. The other two sections in Table 8.1, indicate the logic involved in deciding the type of impact (indirect, secondary and non-cautionary) based on the type (phi or an alpha) of the changing element.

In addition to the above-mentioned factors, it is also vital to watch that a particular constraint (phi) or action (alpha) are not placed in two categories of impacts. The way to resolve this is by ranking the impacts. The impacts are ranked in the order of *direct*, *indirect*, *secondary* and *non-cautionary*; where *direct* has the highest impact and *non-cautionary* has the lowest impact. If an element is found to be in two categories, it is placed in the higher impact category.

There are a number of places in the table which has (-) entries. These indicate the situations where it is logically not possible for sibling elements to exist based on directly impacted node and parent node. To decide this, the set of invariants that were previously introduced in Chapter-6 are used.

The C++ code that implements this logic in Table 8.1 and detailed puchedo code for the implementation is presented, respectively in Appendix-A and Appendix-B of this thesis.
Table 8.1- Decision Table for the Algorithmic-Step-3

<table>
<thead>
<tr>
<th>Parent</th>
<th>IN EACH STEP BACK UP TO THE ROOT (ALGORITHMIC-STEP-2)</th>
<th>When Direct Impacted Node is a Phi</th>
<th>When Direct impacted node is an Alpha</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LEFT Brach of the parent node – Impact on</td>
<td>RIGHT Brach of the Parent Node – Impact on</td>
<td>LEFT Brach of the parent node Impact on</td>
</tr>
<tr>
<td></td>
<td>Sibling</td>
<td>Sibling’s Children</td>
<td>Sibling</td>
</tr>
<tr>
<td>Dot (*)</td>
<td>alpha (α)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>phi (φ)</td>
<td>secondary</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Dot (*)</td>
<td>-</td>
<td>non-cautionary</td>
</tr>
<tr>
<td></td>
<td>Plus(+)</td>
<td>-</td>
<td>non-cautionary</td>
</tr>
<tr>
<td></td>
<td>Star (*)</td>
<td>-</td>
<td>non-cautionary</td>
</tr>
<tr>
<td></td>
<td>Negate(-)</td>
<td>-</td>
<td>Phi – non-cautionary</td>
</tr>
<tr>
<td>Plus(+)</td>
<td>alpha (α)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>phi (φ)</td>
<td>non-cautionary</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Dot (*)</td>
<td>-</td>
<td>Phi – non-cautionary</td>
</tr>
<tr>
<td></td>
<td>Plus(+)</td>
<td>-</td>
<td>Phi – non-cautionary</td>
</tr>
<tr>
<td></td>
<td>Star (*)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Negate(-)</td>
<td>-</td>
<td>Phi – non-cautionary</td>
</tr>
<tr>
<td>Star(*)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Negate(-)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>phi (φ)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>alpha (α)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

In this table the dash (-) indicates the situations where it is logically not possible for sibling elements to exist based on the type of directly impacted node and parent node.
In summary, a mechanism was presented for locating DISN impacts using some entities and KAT expressions captured in the **PECIA Model** (Figure 8.2). In particular, an algorithm and the associated decision table (Table 8.1) were presented, to facilitate locating *indirect, secondary, and non-cautionary* impacts on process elements.

### 8.6 Application of the PECIA Model

In order to exemplify the use of the **PECIA Model**, two evolution scenarios are considered. These evolution scenarios are taken from the *New Course Approval Process* automated within OCAS (OCAS was discussed in detail previously in Chapter-4). Two evolution scenarios are as follows:

- **Evolution Scenario 1**: Recent changes to management organisational structure of University of Western Sydney (UWS) resulted in *removing course officer role* and assigning associated duties to *data and course management officers*.

- **Evolution Scenario 2**: Due to certain operational changes in the organisation, two of the sign-off actions: *Academic registrar systems sign-off* and *Academic registrar operations sign-off* are required to be performed in a single action.

Tables 8.2 and 8.3 present use of the **PECIA Model** to handle the above-mentioned evolution scenarios 1 and 2 respectively. These tables display relevant steps depending on evolution initiation point of each scenario. In both of these examples, affected *phis* and *alphas* are located by searching the KAT Expression of the *New Course Approval Process*, captured in *KAT Expression 6.10* in Chapter-6.
The evolution scenario 1 reflects a change that initiate at the pragmatic level of the PoPA framework.

Table 8.2- Using the PECIA Model to handle Evolution Scenario 1 of the New Course Approval Process of OCAS

<table>
<thead>
<tr>
<th>Steps in the Process using PECIA Model</th>
<th>Results Prompted by the PECIA Model</th>
<th>Process Manager and/or Business Analysts Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step-1 to Step-5 (first iteration)</td>
<td>The courses approval policy and a number of fragments that refer to roles from the UWS Management Structure would be prompted for possible changes.</td>
<td>Out of the long list provided, after careful analysis the process managers would decide to change the fragments 14, 27, 32, 40, 46, 48, 79, 87 and 94 in the Courses approvals policy.</td>
</tr>
<tr>
<td>Step-1 to Step-5 (second iteration)</td>
<td>As it was decided to carryout further contextual artefacts the second iteration would list a set of fragments that are further associated with the once changed in the previous iteration</td>
<td>Process Managers decide no further changes are required to other fragments.</td>
</tr>
<tr>
<td>Step-6 to Step-7 (first iteration)</td>
<td>Based on the changed Fragments of the previous steps the constraints φ₁, φ₉, φ₁₀, φ₁₁, φ₁₂, φ₁₃, φ₁₆, φ₁₇, φ₁₈ and φ₁₉ are identified to be affected (refer Table 6.3 in Chapter-6 where these correlations are defined)</td>
<td>After careful inspection on these selection constraints, the φ₁₆ is selected for the change, as it is the only constraint that links to the course officer change.</td>
</tr>
</tbody>
</table>
| Step-8 to Step-11 (first iteration)  | Based on changing constraint φ₁₆ and the type of change to be identified as a participant modification of the New Course Approval Process (KAT Expression 6.10 in Chapter-6 captured into a binary-tree) is analysed to find the DISN impacts of the change. Algorithm provides the following findings:  
  **Directly affected elements** – φ₁₆  
  **Indirectly affected elements**  - φ₄₆φ₇₂ φ₁₁₅α₁₃  
  **Elements with secondary impacts**  - The list of Phis (φ₁₇, φ₁₈, φ₁₉, φ₂₀, φ₂₁, φ₂₂, φ₂₃, φ₂₄, φ₂₅, φ₂₆, φ₂₇, φ₂₈, φ₂₉, φ₃₀, φ₃₁, φ₃₂, φ₃₃, φ₃₄, φ₃₅, φ₃₆, φ₃₇, φ₃₈, φ₃₉, φ₄₀, φ₄₁, φ₄₂, φ₄₃, φ₄₄, φ₄₅, φ₄₆, φ₄₇, φ₄₈, φ₄₉, φ₅₀, φ₅₁, φ₅₂, φ₅₃, φ₅₄, φ₅₅, φ₅₆, φ₅₇, φ₅₈, φ₅₉, φ₆₀, φ₆₁, φ₆₂, φ₆₃, φ₆₄, φ₆₅, φ₆₆, φ₆₇, φ₆₈, φ₆₉, φ₇₀, φ₇₁, φ₇₂, φ₇₃, φ₇₄, φ₇₅, φ₇₆, φ₇₇, φ₇₈, φ₇₉, φ₈₀, φ₈₁, φ₈₂, φ₈₃, φ₈₄, φ₈₅, φ₈₆, φ₈₇, φ₈₈, φ₈₉, φ₉₀, φ₉₁, φ₉₂, φ₉₃, φ₉₄, φ₉₅, φ₉₆, φ₉₇, φ₉₈, φ₉₉, φ₁₀₀, φ₁₀₁, φ₁₀₂, φ₁₀₃, φ₁₀₄, φ₁₀₅, φ₁₀₆, φ₁₀₇, φ₁₀₈, φ₁₁₄, φ₁₁₅, φ₁₁₆, φ₁₁₈)  
  The list of alphas  – (α₁₃, α₁₄, α₁₅, α₁₆, α₁₇, α₁₈, α₁₉, α₂₀, α₂₁, α₂₂, α₂₃, α₂₄, α₂₅, α₂₆, α₂₇, α₂₈, α₂₉, α₃₀, α₃₁, α₃₂, α₃₃)  
  **Elements with non-cautionary Impacts**  - The list of phis (φ₁, φ₂, φ₃, φ₄, φ₅, φ₆, φ₇, φ₈, φ₉, φ₁₀, φ₁₁, φ₁₂, φ₁₃, φ₁₄, φ₁₅, φ₁₆, φ₁₇, φ₁₈, φ₁₉, φ₂₀, φ₂₁, φ₂₂, φ₂₃, φ₂₄, φ₂₅, φ₂₆, φ₂₇, φ₂₈, φ₂₉, φ₃₀, φ₃₁, φ₃₂, φ₃₃, φ₃₄, φ₃₅, φ₃₆, φ₃₇, φ₃₈, φ₃₉, φ₄₀, φ₄₁, φ₄₂, φ₄₃, φ₄₄, φ₄₅, φ₄₆, φ₄₇, φ₄₈, φ₄₉, φ₅₀, φ₅₁, φ₅₂, φ₅₃, φ₅₄, φ₅₅, φ₅₆, φ₅₇, φ₅₈, φ₅₉, φ₆₀, φ₆₁, φ₆₂, φ₆₃, φ₆₄, φ₆₅, φ₆₆, φ₆₇, φ₆₈, φ₆₉, φ₇₀, φ₇₁, φ₇₂, φ₇₃, φ₇₄, φ₇₅, φ₇₆, φ₇₇, φ₇₈, φ₇₉, φ₈₀, φ₈₁, φ₈₂, φ₈₃, φ₈₄, φ₈₅, φ₈₆, φ₈₇, φ₈₈, φ₈₉, φ₉₀, φ₉₁, φ₉₂, φ₉₃, φ₉₄, φ₉₅, φ₉₆, φ₉₇, φ₉₈, φ₉₉, φ₁₀₀, φ₁₀₁, φ₁₀₂, φ₁₀₃, φ₁₀₄, φ₁₀₅, φ₁₀₆, φ₁₀₇, φ₁₀₈, φ₁₁₄, φ₁₁₅, φ₁₁₆, φ₁₁₈)  
  The list of alphas (α₁, α₂, α₃, α₄, α₅, α₆, α₇, α₈, α₉, α₁₀, α₁₁) | Since the directly affected element φ₁₆ was previously changed and no other elements are directly affected further changes are not required. |
<p>| Step-12 to Step-13 | Due to the change in the previous steps, the Process Models and Participant Models of OCAS | The appropriate changes are carried out in |</p>
<table>
<thead>
<tr>
<th>Steps in the Process using PECIA Model</th>
<th>Results Prompted by the PECIA Model</th>
<th>Process Manager and/or Business Analysts Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>(first iteration)</td>
<td>implementation are identified to be affected.</td>
<td>Process Models and Participant Models</td>
</tr>
<tr>
<td>Step-14 to Step-17 (first iteration)</td>
<td>Due to the changes carried out in Process Models and Participant Models, the User Interface Definition model is identified to be affected.</td>
<td>Decided to change the User interface Definition Model</td>
</tr>
<tr>
<td>Step-14 to Step-17 (second iteration)</td>
<td>No more models are identified to be affected due to the previous User interface Definition Model change; therefore move to the next step.</td>
<td></td>
</tr>
<tr>
<td>Step-18 to Step-19 (first iteration)</td>
<td>Due to the model changes carried out in the previous step the configuration file ocas_nc.wfi and data store named ocas_instance are identified as being affected.</td>
<td>The configuration file ocas_nc.wfi and data store named ocas_instance are changed according to the requirements</td>
</tr>
<tr>
<td>Step-20 to Step-23 (first iteration)</td>
<td>Due to changes in configuration file ocas_nc.wfi and data store named ocas_instance, the presentation template named enter_course.tt is identified to further affected</td>
<td>The presentation template enter_course.tt is identified as needing change.</td>
</tr>
<tr>
<td>Step-20 to Step-23 (second iteration)</td>
<td>Due to the template enter_course.tt none of the other implementation artefacts are identified to be affected.</td>
<td></td>
</tr>
</tbody>
</table>
| Step-24 to Step-27                   | • Appropriate tests were carried to measure the success of the above-mentioned change and proved the success of the changes.  
• The method of handling process instance transfers is to use the flush approach, were the existing instances are allowed to proceed with old process definition | |
The evolution scenario 2 reflects a change that initiates at the semantic level of the PoPA framework.

### Table 8.3- Using the PECIA Model to handle Evolution Scenario 2 of the New Course Approval Process of OCAS

<table>
<thead>
<tr>
<th>Steps in the Process using PECIA Model</th>
<th>Results Prompted by the PECIA Model</th>
<th>Process Manager and/or Business Analysts Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step-7 to Step-8</strong> (first iteration)</td>
<td>It is identified that the action ( \alpha_{23} ) — “Office of Academic Registrar–System’s (OAR-Sys) sign-off” is removed and action ( \alpha_{23} ) — “Office of Academic Registrar–Operation’s (OAR-opr) sign-off” encapsulate the tasks performed by the removed action.</td>
<td>Make a temporary record of these changes</td>
</tr>
</tbody>
</table>
| **Step-9 to Step-11** (first iteration) | Based on changing actions and the type of changes the following are identified to be the DISN impacts on the New Course Approval Process in OCAS (KAT Expression 6.10 in Chapter-6 captured into a binary-tree):  
**Directly affected elements** –  
The list of phis - \( \phi_{106} \) (which waits for the completion of action \( \alpha_{23} \)), \( \phi_{37} \) (that associates the role assistant_registrar_operations to the action), \( \phi_{90} \) (which defines the visibility of data in the removed action), \( \phi_{95} \) (which defines an integrity constraint related to the removed action)  
The list of alphas - \( \alpha_{18}, \alpha_{25} \)  
**Indirectly affected elements** –  
The list of phis - \( \phi_{22}, \phi_{29}, \phi_{55}, \phi_{62}, \phi_{79}, \phi_{88}, \phi_{100}, \phi_{101}, \phi_{102}, \phi_{103}, \phi_{104}, \phi_{105}, \phi_{107} \)  
The list of alphas - \( \alpha_{24}, \alpha_{27}, \alpha_{28}, \alpha_{29}, \alpha_{30}, \alpha_{32} \)  
**Elements with secondary impacts** –  
The list of phis - \( \phi_{23}, \phi_{24}, \phi_{25}, \phi_{26}, \phi_{27}, \phi_{28}, \phi_{29}, \phi_{30}, \phi_{31}, \phi_{32} \)  
The list of alphas - \( \alpha_{19}, \alpha_{20}, \alpha_{21}, \alpha_{22}, \alpha_{23}, \alpha_{24} \)  
**Elements with non-cautionary Impacts** –  
The list of phis - \( \phi_{1}, \phi_{2}, \phi_{3}, \phi_{4}, \phi_{5}, \phi_{6}, \phi_{7}, \phi_{8}, \phi_{16}, \phi_{17}, \phi_{18}, \phi_{19}, \phi_{20}, \phi_{21}, \phi_{27}, \phi_{38}, \phi_{19}, \phi_{40}, \phi_{41}, \phi_{42}, \phi_{43}, \phi_{44}, \phi_{45}, \phi_{51}, \phi_{52}, \phi_{53}, \phi_{54}, \phi_{70}, \phi_{71}, \phi_{72}, \phi_{73}, \phi_{74}, \phi_{75}, \phi_{77}, \phi_{78}, \phi_{113}, \phi_{115}, \phi_{116}, \phi_{119} \)  
The list of Alphas - \( \alpha_{1}, \alpha_{2}, \alpha_{3}, \alpha_{4}, \alpha_{5}, \alpha_{6}, \alpha_{7}, \alpha_{12}, \alpha_{15}, \alpha_{16}, \alpha_{17}, \alpha_{33} \) | The following changes are decided on the directly affected elements  
\( \phi_{106} \) – Remove as this condition does not need to be checked any more  
\( \phi_{37} \) – Remove as this condition does not need to be checked any more  
\( \phi_{90} \) – Add to the composite guard condition of action \( \alpha_{23} \)  
\( \phi_{95} \) - Add to the composite guard condition of action \( \alpha_{23} \) |

**Step-12 to Step-13** (first iteration)  
Due to the change in the previous steps, the Process Models and Participant Models of OCAS implementation are identified to be affected.  
The appropriate changes are carried out in Process Models and Participant Models

**Step-14 to Step-17**  
Due to the changes carried out in Process Models and Participant Models, the User Interface Definition  
Decided to change the User interface
<table>
<thead>
<tr>
<th>Steps in the Process using PECIA Model</th>
<th>Results Prompted by the PECIA Model</th>
<th>Process Manager and/or Business Analysts Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>(first iteration) model is identified to be affected</td>
<td></td>
<td>Definition Model</td>
</tr>
<tr>
<td>Step-14 to Step-17 (second iteration) No more models are identified to be affected due to the previous User interface Definition Model change; therefore move to the next step.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step-18 to Step-19 (first iteration) Due to the model changes carried out in the previous step the configuration file ocas_nc.wfi and data store named ocas_instnace are identified as being affected.</td>
<td>The configuration file ocas_nc.wfi and data store named ocas_instnace are changed according to the requirements</td>
<td></td>
</tr>
<tr>
<td>Step-20 to Step-23 (first iteration) Due to changes in configuration file ocas_nc.wfi and data store named ocas_instnace, the presentation template named enter_course.tt is identified to be affected.</td>
<td>The presentation template enter_course.tt is decided for changes</td>
<td></td>
</tr>
<tr>
<td>Step-20 to Step-23 (second iteration) Due to changes to the template enter_course.tt none of the other implementation artefacts are identified to be affected.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Step-24 to Step-27 | • Appropriate tests were carried to measure the success of the above-mentioned change and proved the success of the changes.  
• The method of handling process instance transfers is to use the flush approach, were the existing instances are allowed to proceed with old process definition | |
This section exemplified the use of the **PECIA Model** to handle two evolution scenarios related to the *New Course Approval Process* automated within OCAS.

### 8.7 Chapter Summary

This chapter presented two core contributions of this thesis: the **PECIA Model** and the algorithm that locates the DISN impacts.

The **PECIA Model** captures data structures and schemata previously developed in Chapter-5 and 6 into a single model. In addition, this model also encapsulates the *nature of process evolution*, discussed in Chapter-7, in order to facilitate reflecting high-level changes, effectively and efficiently in web-workflow artefacts. The use of the **PECIA Model** is discussed in detail. This discussion was organised into four sections to demonstrate the ability of the **PECIA Model** to handle changes that can initiate at four levels (*pragmatic, semantic, syntactic, and implementation*) of the PoPA framework.

The second contribution is an algorithm that could search the KAT expressions to locate the propagating impact. The in-depth logic involves deciding DISN impact of process element changes is described using a decision table.

Both the **PECIA Model** and the analytical algorithm have prototypic implementations. Using this basic implementation and certain evolution scenarios drawn from the OCAS case study, the validity of both the **PECIA Model** and algorithm is assessed.

The practical utility of the **PECIA Model** was highlighted in this chapter. Next chapter focuses on validating both practical and epistemic utility of all contributions of this research.
Chapter-9

“It is not the mountain we conquer but ourselves.”
Sir Edmund Hillary

“The important thing about a problem is not its solution, but the strength we gain in finding the solution.”
Author Unknown

9 Discussion, Conclusion and Future Research Directions

9.1 Chapter Overview

This final chapter is predominantly aimed at highlighting the contributions of this research and validating both practical and epistemic utility of those contributions. This thesis provides a holistic approach for the problem of managing business process evolutions and changes in web-based workflow systems. The core solution provided for the problem is Process Evolution and Change Impact Analysis (PECIA) Model, which captures meta-information of process and its automation. This model facilitates locating the propagating impact of evolutions that could get initiated at various places down to the implementation artefacts of the web-based workflow system.

This chapter is organised as follows. Initially, this chapter summarises the research presented in this thesis. Secondly, the contributions of this research are identified. Then proof of practical utility of these contributions carried out in various places of this thesis is summarised. Followed by this, epistemic utility of the contributions are discussed based on the theoretical framework presented in Chapter-1. Finally, this thesis is concluded by highlighting a number of possible future research directions.
9.2 Research Summary

The primary motivation for this research came from a real-life process automation activity carried out in a tertiary education institution in Australia. In 2005, Online Course Approval System (OCAS) project was initiated to automate some internal processes at the University of Western Sydney – UWS. In OCAS, relatively large and complex business processes were automated. Identifying the evolutionary nature of the processes, OCAS was implemented with a number of flexibility features (Detailed information in relation to OCAS and its flexibility was discussed in Chapter-4). Despite the adaptable nature of OCAS, when it came to real process evolutions, the process managers were unable to use these features to the full potential; due to their lack of understanding of the associations between the changing process elements and implementation artefacts. This highlighted the need for a repository that could capture meta-information related to a process automation activity.

Due to the problem that was outlined above, the following was set as the main question to be addressed in this research; “How can business process evolutions be accurately and effectively reflected in already implemented web-based workflow systems?”

When finding answers to this question, it required a holistic solution, that could locate change impact propagation from contextual artefacts (policies, guidelines, organisational structures, Laws, regulations, and Acts) down to the implementation artefacts of the web-based workflow systems. Therefore, to comprehend this holistic nature of the solution and to organise the discussion a framework named paradigm of process automation – PoPA framework (described in detail in Chapter-1) was introduced. The PoPA framework has four levels as pragmatic, semantic, syntactic, and implementation, which represented the contextual artefacts, operational business process, conceptualised models, and web workflow artefacts respectively.

Careful and thorough literature analysis, carried out in Chapter-3, revealed a lack of research in the area of ‘impact analysis of process evolutions’. Particularly scarcity was evident in research studies that provide holistic solutions, which could handle changes from the pragmatic level down to the implementation level.
A holistic solution needed to facilitate managing changes that could initiate at any of the above-mentioned levels down to the implementation artefacts of web-workflow systems. Therefore, it was vital to record all types of associations that existed among the elements at all four levels of the PoPA framework. These correlations were categorised into two: intra-level and inter-level CAD (Constraints, Associations, and Dependencies). As such, Chapters-5 and 6 collectively contributed to comprehending and finding data structures that could facilitate recording both intra and inter-level CAD.

Intra-level CAD at semantic level had a number of complexities that challenged the use of relational data structures for capturing the CAD among process elements. Therefore, a new algebraic formalism is developed. This was presented in Chapter-6. This formalism was based on Kleene Algebra with Test (KAT), which is a bi-partite algebra that allows creating linear process expressions for analytical purposes, requires for this research. These linear KAT expressions were mapped into binary-tree structures for implementation.

The success of BPEM in web workflows, not only depends on comprehending the correlations among elements at different levels, but also the nature of evolution of process elements. As such, in Chapter-7 a detail discussion was carried out to understand the primitives, semantic and dynamics of process evolution. Based on these primitives, semantics and dynamics, four types of impacts were identified as direct, indirect, secondary and non-cautionary (DISN).

Chapter-8 of this thesis brought together all the concepts, data structures, and theories developed in previous chapters, into Process Evolution and Change Impact Analysis (PECIA) Model. This model has a hybrid structure that captures both intra and inter-level CAD among elements in relational structures and binary-trees in a single model.

As indicated in the research methodology chapter (Chapter-2), this research mainly used the constructive research approach. In constructivism, there are mainly five stages: i) problem definition, ii) understanding theoretical body of knowledge, iii) construction of the solution, iv) validating practical utility of the solution, and v) validating epistemic applicability of the solution. Accordingly, the thesis was organised as follows:

• Problem definition
  o Chapter-1(Introduction) and
9.3 Contributions of this Research

A number of contributions arise from this work. Some of the contributions are core and mainly geared towards facilitating effective management of process evolutions in web-workflow systems. There are other secondary contributions, which are used to set the background to this work. However, these secondary contributions are vital and could be used in different other works.

The contributions of this research are presented under these two categories: primary and secondary contributions.

9.3.1 Primary Contributions

There are mainly three primary contributions of this work as: a) Use of KAT to capture CAD among process elements, b) the PECIA Model, and c) Analytical algorithm locates DISN impacts of process evolutions using linear KAT expressions.
• **Use of KAT to capture CAD among process elements:** This thesis provides a novel approach to capture correlations among process elements using KAT. KAT allowed creating linear process expressions, which are suitable for computer manipulation. Further, bi-partite nature of KAT allows capturing all types of associations among process elements. In order to capture CAD among process elements 21 KAT definitions are introduced in Chapter-6.

• **PECIA Model:** This model cohesively and completely captures both *intra* and *inter-level CAD* among elements in the *PoPA framework*, into a single framework. This model and its usage are discussed in detail in Chapter-8. There are mainly three layers in the **PECIA Model**. First layer named, *Element Registries* captures the *intra-level associations* among elements in the four levels: *pragmatic*, *semantic*, *syntactic*, and *implementation*. It is worth noting that these registries record only references to contextual artefacts, models, and system component parts, as opposed to synthesising these external elements. In any case, if a particular organisation is already using a sophisticated system that interprets contextual artefacts, this also can be supported by the registries in the **PECIA Model**. The second layer named *Element Mapping* captures all *inter-level CAD among four levels*. The third layer named as *Evolution Mapping* layer facilitates recording other important information that is required in process evolutions.

• **Analytical algorithm that locates DISN impacts of process evolutions using linear KAT expressions:** The third main contribution of this work is an analytical algorithm developed to identify DISN impacts by searching the KAT expressions in binary-tree structures. As it was mentioned previously in Chapter-8, the depth of a binary-tree, that represents KAT expressions, is an aspect that is non-determinable priori design. The depth depends on the complexity of the process. As such, it requires an algorithm that could analyse process of any complexity (or binary tree of any depth). The analytical algorithm is comprehensive as it considers a number of factors in deciding the nature of impacts on process elements. The core logic embedded in the algorithm is illustrated in a decision table and has a prototype implementation in C++.
9.3.2 Secondary Contributions

There are four secondary contributions of this research: a) The PoPA framework, b) The CAD Model of Process Elements, c) Representation of linear KAT expressions as binary-tree structures for implementation purposes, and d) Taxonomy of process elements evolutions based on primitives, semantics, and dynamics of change. These contributions either set the background or facilitate the primary contributions. They are summarised below.

- **The PoPA framework**: This framework has four levels of representations of the process as pragmatic, semantic, syntactic and implementation. While this framework was not an exclusively new concept, it has certain novel ideas added to previous models to make it usable within this research. These include the concept of leakage of information when process representations move from one level to another. Further, this framework served as a major influence in structuring this research and providing a holistic solution to the problem.

- **The CAD Model of Process Elements**: The CAD model of Process Elements captures the correlations among process elements into a diagrammatic model (Figure 5.18 in Chapter-5). This model is similar to a meta-model that depicts all possible associations among process elements. As such, this model is the basis for defining the 21 KAT definitions, which are used to create linear expressions of process elements.

- **Representation of linear KAT expressions as binary-tree structures for implementation purposes**: While the representation of linear expressions in binary-tree is a concept that has been used over three decades (Shave, 1975), its usage still proves to be correct. Based on the core principles of binary-trees, Chapter-6 of this thesis presents a set of invariants that facilitate capturing a process into KAT based linear expressions without breaching its syntax or semantics.

- **Taxonomy of process elements evolutions based on primitives, semantics, and dynamics of evolution**: Chapter-7 carries out a comprehensive study to reveal the nature of process element evolutions. In this study the primitives,
semantics, and dynamics of process element evolution are discussed in relation to all process elements—actions, participants, and business object. While this study into process element evolution is not novel, it gives a different and practical perspective, which is beyond workflow modelling, to study process evolutions.

9.4 Practical Relevance

In constructive research approach, practical relevance refers to validating the use of a particular concept, model or theory, to solve real-life problems in a given domain (Lukka, 2000). In this view, this research extensively uses the New Course Approval Process automated in OCAS as the case study to validate the contributions. The New Course Approval Process automation within OCAS, is explained in detail previously in Chapter-4.

The practical relevance of the contributions is validated in the chapters where these are introduced, and summarised in Table 9.1 below.

<table>
<thead>
<tr>
<th>Contribution</th>
<th>Practical Relevance Validated in</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use of KAT for Process Modelling</td>
<td>The practical use of KAT for modelling the New Course Approval Process is demonstrated in Chapter-6. The complete KAT expression is gradually developed based on the KAT definitions that are introduced in the chapter. The full New Course Approval Process is presented in KAT Expression 6.10 in Chapter-6.</td>
</tr>
<tr>
<td>PECIA Model</td>
<td>The practical utility of this model was presented in Chapter-8, when it was used for handling two evolutions scenarios in relation to OCAS.</td>
</tr>
<tr>
<td>Analytical Algorithm to Analyse KAT based binary trees for impact analysis</td>
<td>The practical use of analytical algorithm was also proven in Chapter-8, when it was used in solving the evolutions scenarios of OCAS.</td>
</tr>
<tr>
<td>PoPA framework</td>
<td>This framework was proven successful for organising workflow related research, as it was extensively used to structure this research. In particular, when presenting the case study in Chapter-4, and developing the solutions in Chapter-5 and 6, the PoPA framework was extensively used.</td>
</tr>
<tr>
<td>CAD Model of Process elements</td>
<td>The use of this model as a meta-model that capture correlations among process elements were proven based on its usage to create the KAT definitions.</td>
</tr>
</tbody>
</table>
Contribution | Practical Relevance Validated in
--- | ---
Representation of KAT expressions in binary-tree structure | The applicability of this approach was proven by using these data structures to search the KAT expressions successfully for analytical purposes in Chapter-8.

| Taxonomy of process evolutions | The use of the taxonomy was presented in practical scenario in Chapter-8, when the PECIA Model was used to identify the propagating impact of process element evolutions in OCAS related scenarios.

The primary contributions of this research (listed in Table 9.1) support the organisations to carryout process evolutions in already implemented web workflow systems both accurately and effectively. Generally, organisations have to make significant investments to reflect process evolutions that take place after the implementation of workflow systems. As a solution to this, workflow systems now abstract the process logic into machine-readable process definitions. However, making changes to the process definitions (via visual tools or directly) relies on the analytical capability of the humans who carryout the task. The task of making processes changes by humans can be rather challenging: when processes are large and complex and people who carryout the changes are not familiar with the inner workings of the process. Therefore, the PECIA model introduced in this research provides a tool for humans to carryout process evolution tasks accurately, reducing the downtime of workflow systems and with minimum costs.

As summarised in Table 9.1, the applicability of the contributions is proved using OCAS case study. The only limitation of this approach is that OCAS being the only case study used to validate the contributions. Although OCAS is considered a success story in process automation, validation of these contributions with a few other practical process automation experiences would have strengthened its practical utility.

**9.5 Theoretical Relevance**

The theoretical framework, within which this research is placed, is presented in Chapter-1 (Figure 1.2 in page 13) of this thesis. In addition, the literature review presented in Chapter-3 of this thesis is structured around this theoretical framework.

In this section, the same theoretical framework is used to indicate the applicability of the contributions of this research in the areas of BPA and BPEM. In
this view, Figure 9.1 illustrates the applicability of both primary and secondary contributions of this research within the main theoretical framework.

As indicated in Figure 9.1, the primary contributions **PECIA Model** and **KAT Expression Analysis Algorithm** directly contribute to ‘impact analysis of business process evolution’, which is identified as the focus of this research. The other primary contribution - *use of KAT for Process Modelling*, is identified to contribute to the **PECIA Model**.

The secondary contributions support this research in varying degrees. The **PoPA framework** allowed structuring this research and devising a solution. Therefore, this model contributes to the broader areas of BPA and BPEM as indicated in Figure 9.1. The **CAD model of process elements** served as a meta-model in creating KAT definitions (21 KAT based definitions are introduced in Chapter-6) to capture the correlations among process elements. The taxonomy of process evolutions-*primitives, semantics and dynamics*, supports finding the propagating impact using the **PECIA Model**. Finally **binary tree representation of KAT expressions** facilitated the implementation.

### 9.5.1 PECIA Model viewed as a Knowledge Repository

While the **PECIA Model** does not comply with the full set of characteristics, for it to be considered as a knowledge repository, it can be reasoned that this model
is a knowledge repository in its infancy. In order to understand this aspect, some information in relation to knowledge repositories is presented.

In an organisational context, knowledge is defined as information that is combined with experience, context, interpretation, and reflection (Davenport et al., 1997). When this information is collected into a single location for data mining and analytical purposes it is considered to be a knowledge repository (Yoshioka et al., 2001). Based on this view, collecting information in relation to associations among process elements and recording changes of evolutions suggests the **PECIA Model** to be considered a knowledge repository, especially supporting the evolution of processes. However, only one analytical aspect is supported by the **PECIA Model** and that is evolution impact analysis. In other words, in the present form the **PECIA Model** does not attempt to support sophisticated data mining concepts to discover new knowledge. However, these other analytical uses of the **PECIA Model** can be considered as a future research directions of this model.

### 9.5.2 Comparison with Similar Work

Some of the contributions of this thesis are developed based on the existing theoretical body of knowledge to achieve the objectives of this research. Therefore, these concepts do not compete with previous similar work, but contribute to the theoretical body of knowledge in the area of BPA and BPEM.

Table 9.2 below presents the nature of association between the contributions of this work and previous similar work.

**Table 9.2- Comparison of the Contributions of this Research with Similar Work**

<table>
<thead>
<tr>
<th>Primary Contributions</th>
<th>Similar or associated work</th>
<th>How this contribution is different or complement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use of KAT for Process Modelling</td>
<td>Petri-Nets (van der Aalst et al., 1994; Murata, 1989; Reisig, 1985) and Process Algebra (Basten, 1998; Fokkink et al., 1994; van Glabbeek, 1987) are two formalisms that are commonly used for process modelling.</td>
<td>The advantages of the use of KAT are explained in Chapter-6. Further, the section 9.4.1.1 below solves a challenge set by van der Aalst (2003) in using formalisms for process modelling.</td>
</tr>
<tr>
<td>PECIA Model</td>
<td>There is only few research works (Ramesh et al., 2005; Soffer, 2005; Bodhuin et al., 2004; Min et al., 1999) that identify the need for impact analysis.</td>
<td>None of the previous works provides a holistic approach that spans from the pragmatic level down to the implementation artefacts and addresses the evolution of all process dimensions (actions,</td>
</tr>
</tbody>
</table>
### Contribution

<table>
<thead>
<tr>
<th>Contribution</th>
<th>Similar or associated work</th>
<th>How this contribution is different or complement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytical Algorithm to Analyse KAT based binary trees for impact analysis</td>
<td>The other impact analysis methods are suggested by Soffer (2005) and Ramesh et al. (2005). Soffer (2005) suggests impact analysis based of set of guidelines. Ramesh et al. (2005) uses a knowledge management system that uses inference based mechanism.</td>
<td>This is a specific algorithm developed for use in this research.</td>
</tr>
<tr>
<td>PoPA framework</td>
<td>This was developed based on previous models by zur Muehlen (2001) and Georgakopoulos et al. (1995).</td>
<td>Complements and extends these previous models</td>
</tr>
<tr>
<td>CAD Model of Process elements</td>
<td>There were a number of meta-models (Russell et al., 2006b; Russell et al., 2006a; Russell et al., 2005a; Russell et al., 2005b; van der Aalst et al., 2003b; Kappel et al., 2000; WfMC, 1999; Kwan et al., 1997; Sadiq et al., 1997; Gottlob et al., 1996; Eder et al., 1995; Kappel et al., 1995; Bussler et al., 1995; Eder et al., 1994) discussed previously in Chapter-3 (literature review)</td>
<td>This is a specific implementation used for this research, which proves the applicability of these approaches.</td>
</tr>
<tr>
<td>Representation of KAT expressions in binary-tree structure</td>
<td>Based on the previous works of Shave (1975) and Knuth (1973)</td>
<td></td>
</tr>
<tr>
<td>Taxonomy of process evolutions</td>
<td>Based on the previous works Soffer, van der Aalst and Jablonski (2000), Casati, Ceri, Pernici, and Pozzi (1998), and Sadiq et al. (1997)</td>
<td>These primitives, semantics, and dynamics were introduced based on a number of previous works. However, the element evolution dynamics gives a new perspective towards understanding the evolutionary nature of processes.</td>
</tr>
</tbody>
</table>

### 9.5.2.1 Solution to a Process Modelling Challenge using KAT

In the paper by van der Aalst (2003), which counter argues the use of Process Algebra for process modelling as opposed to Petri-Nets, he sets a challenge to model the process shown in Figure 9.2.
Due to the uni-partite nature of Process Algebra the above process scenario is not possible to be modelled using Process Algebra, particularly to handle the flow of action from c to f (this is explained by van der Aalst (2003)). This same process model is used to show the theoretical relevance of KAT for process modelling.

According to the KAT definition-1 (given in Chapter-6), the alphas associated with the above process are identified as $\alpha_a$, $\alpha_b$, $\alpha_c$, $\alpha_d$, $\alpha_e$, $\alpha_f$, $\alpha_g$ and $\alpha_h$. Then based on KAT definition-2 the relevant phis are identified as $\phi_a$, $\phi_b$, $\phi_c$, $\phi_d$, $\phi_e$, $\phi_f$, $\phi_g$ and $\phi_h$. Using the KAT Definitions-4 to 8, the process in Figure 9.2 can be modelled into a linear KAT expression as follows:

\[
(\phi_d \alpha_d) \phi_m((\phi_h \alpha_h)(\phi_c \alpha_c)((\phi_d \alpha_d)+(\phi_f \alpha_f)(\phi_g \alpha_g))+(\phi_e \alpha_e)(\phi_f \alpha_f)(\phi_g \alpha_g))^*(\phi_h \alpha_h)
\]

Phi element identified as $\phi_m$ is a special guard elements that support parallel construct as explained in KAT Definition-7 in Chapter-6. In this expression, however the actions $\alpha_f$ and $\alpha_g$ appears twice, the enactment of those actions can be restricted to once using the composite guard conditions $\phi_f$ and $\phi_g$.

This simple exercise shows the powerfullness of bi-partite representation of KAT (similar to Petri-Nets) in creating linear process expressions, suitable for computer manipulation. This further proves the theoretical relevance of KAT for process modelling purposes.

### 9.6 Limitations

Although this work has proven to be useful from both practical and theoretical points of view, there are a few limitations in this work. The primary cause for these limitations is the time constraints, which force certain milestones such as completion of thesis writing, to be achieved before deadlines. However, the
identified limitations of this research are presented below to indicate possible continuous improvement of this work.

At present, it requires the **PECIA Model** to be manually populated by process managers or business analysts. This can be a challenging task depending on the complexity of the process. Therefore, it would be appropriate to have a semi-automated mechanism that could facilitate the **PECIA Model** data population. For example such as reading from existing process models in order to create the KAT expressions.

The present prototypic implementation of the **PECIA Model** requires handling a single change at a time, using the process explained in Chapter-8. However, it would have been useful if this model and the searching mechanism can handle multiple changes at one time. In addition, the present prototypic implementation of the **PECIA Model** can be further tested by linking it with a commercial workflow management system.

As mentioned before the findings of this research are validated using only a single case study based on OCAS. The rationale for usage of a single, yet a complex and large case study was to show the challenging nature of managing processes involved in such case studies. It was decided to provide an end-to-end solution with more focus on key research issues using a single case study rather than, using multiple case studies that would further increase the length of this research and hence diluting the intended focus.

### 9.7 Future Research Directions

Based on the wide range of contributions provided in this research; there are a number of further research directions that can be founded on this work.

The wider applicability of KAT expressions can be explored in a number of ways. For instance, compact KAT expressions of processes could be used to compare similarities or differences between processes. This comparison could be used for evolution analysis. Further, KAT being an algebraic expression, it can be used to investigate possible optimisation opportunities of a process. This optimisation can be achieved by locating redundancy of actions. Further, the use of KAT expressions as executable process expressions is another interesting research direction. Workflow
engines that can synthesis a KAT expression can enact the process according to the flow control recorded there.

In this research, DISN impacts are used only for understanding further evolution needs of the process. However, these impacts can be also used in deciding on process instance transfers also referred to as dynamic evolution, to find the position of certain actions, while the evolutions were taking place.

Here the idea is to use the PECIA Model to facilitate already implemented web-workflow systems. However, it could also be used to guide the initial process automation in a very systematic manner. This different use of the PECIA Model for initial process automation is another research direction that can be further explored.

Further, the PECIA Model can be enhanced for it to be a knowledge repository that could facilitate various sophisticated data mining features. Such an extension to the PECIA Model can be used for enhancing the process evolution task by suggesting the possible changes based on previous changes.

At present, the PECIA Model captures meta-information related to a single process and its automation. However, this model can be easily extended to capture multiple processes and the associations among those processes. Such an extension would facilitate locating propagating impact of not only across processes, but also across organisational boundaries.

9.8 Chapter Summary and Conclusion

This conclusion chapter first summarised the contributions of this research into two categories as primary and secondary. Three primary contributions were outlined as: a) Use of KAT to capture CAD among process elements, b) the PECIA Model, and c) Analytical algorithm that locates DISN impacts of process evolutions using linear KAT expressions. There were four secondary contributions identified: a) PoPA framework, b) The CAD Model of Process Elements, c) Representation of linear KAT expressions as binary-tree structures for implementation purposes, and d) Taxonomy of process elements evolutions based on primitives, semantics, and dynamics of change. These secondary contributions set the background to this research and facilitate achieving the primary contributions.

Then this chapter evaluated the practical and epistemic utility of both primary and secondary contributions of this research. The practical utility of these
contributions was demonstrated in the chapters where these were initially introduced. Hence, Table 9.1 summarised the practical applications that was carried out in different chapters. The epistemic utility of this research was presented using the broader theoretical framework of this research (Figure 9.1). Further, a comparison with important similar literature in both BPA and BPEM areas was carried out and presented in Table 9.2. In order to prove the epistemic utility of using KAT for process modelling further, this section demonstrated the use of KAT, by addressing a process modelling challenge that was set by van der Aalst (2003).

Finally, this chapter presented the limitations of this research in its current form and possible future research directions. The outcomes of this research can be viewed as the beginning of a progression of a number of new research studies such as exploring further uses of KAT based process expressions and the **PECIA Model**.

In conclusion, this research has achieved its overall objective as set out in Chapter-1, by developing a framework; the **PECIA Model** to manage the process evolutions effectively and efficiently in web-based workflow systems.
REFERENCES


Lukka, K. (2000). The key issues of applying the constructive approach to field research: School of Economics and Business Administration.


BIBLIOGRAPHY


APPENDIX A

The following C++ code implements a mechanism of identifying indirect, secondary and non-cautionary impacts on other process elements, based on the directly impacted element. This logic was presented in Table 8.1 in Chapter-8.

```cpp
// The following structure will keep links to parent node, left
//node, right node and the symbol recorded in each node
// symbols will be either a phi, alpha, ., +, *, ~(negation)
struct nodes
{char parent;
 char left;
 char right;
 char symbol;};
// The elementArray will record the whole KAT expression
//This will be pre-populated using in-order binary tree traversal
//method
nodes elementArray[];
//This is the element that is changed. This is either a phi or and
//alpha. This will be given to this algorithm
char changing_element;
//the following struct allows to define a stack type
struct stack
{int depth;// stack size
 int current;// pointer to bottom of stack
 char stack_layer[];}//keeps the actual values
stack indirect_impact;
stack secondary_impact;
stack no_cautionary_impact;
//these two arrays stores the alphas and phis separately of the
//sibling
char siblings_children_phis[];
char siblings_children_alphas[];
//Find the exact position where the changing element is in the array
//that holds the KAT expression
do while
  elementArray[x].symbol != changing_element;
x++;
//Once the changing element is located the immediate the parent
//element of the changing element is accessed. the get_element is a
//function that retrieves the position of the //parent element
int parent= get_parent(elementArray[x]);
//element_is_a_phi is function that returns a true if the passed
//element is a phi else returns a false
case element_is_a_phi(changing_element):
  //if changing element is a phi this is how impact on other
//elements are identified. this loop search backward until it
//reaches //the root or the node value changes something other than
//'.'
  //the node values can be only '.' or '+' the other values (i.e. *,
~, //phi or alpha) are not valid to be in the nodes above a phi
//(which //is the changing element)according to above case statement
  [while (elementArray[parent] != root and
     elementArray[parent].symbol ='.')?>
```

PhD thesis - Change Impact Analysis to Manage Process Evolution in Web Workflows
// the following get_sibling function retrieves the position of the
//sibling node
int sibling = get_sibling(elementArray[x], parent);
// the following get_side_of_sibling function retrieves the branch
//of //the sibling node. this will return either 'right' or 'left'
char side = get_side_of_sibling(elementArray[x], parent);
// get all the phis in the sibling node
siblings_children_phis[] = get_all_phis(elementArray[sibling]);
// get all the alphas in the sibling node
siblings_children_alphas[] = get_all_alphas(elementArray[sibling]);
// if the sibling is on the right side of the parent node
if side == 'right'
// and also if the sibling node is a phi. the element_is_a_phi
//function returns true if the passed element is a phi
{
    if element_is_a_phi(elementArray[sibling])
    {
        secondary_impact.push(elementArray[sibling].symbol);
    }
    Else
    {
        no_cautionary_impact.push(siblings_children_phis[], siblings_children_alphas[]);
    }
} else{
// i.e. if the side of the sibling node is 'right'
// the following case statements checks the node value of the right
// side sibling node and identify the type of impact on alphas and
// phis element_is_a_alpha function returns true if the passed
// element //is a alpha
    case element_is_a_alpha(elementArray[sibling].symbol):
    {
        indirect_impact.push(elementArray[sibling].symbol);
    }
    case element_is_a_phi(elementArray[sibling].symbol):
    {
        secondary_impact.push(elementArray[sibling].symbol);
    }
    case elementArray[sibling].symbol = '.':
    {
        secondary_impact.push(siblings_children_phis[]);
        indirect_impact.push(siblings_children_alphas[]);
    }
    case elementArray[sibling].symbol = '+':
    {
        no_cautionary_impact.push(siblings_children_phis[]);
        indirect_impact.push(siblings_children_alphas[]);
    }
    case elementArray[sibling].symbol = '*':
    {
        no_cautionary_impact.push(siblings_children_phis[]);
        secondary_impact.push(siblings_children_alphas[]);
    }
    case elementArray[sibling].symbol = '~':
    {
        secondary_impact.push(siblings_children_phis[]);
    }
    parent=get_parent(elementArray[prev]);
    }
    case element_is_a_alpha(changing_element):
// if changing element is a alpha this is how impact on other

while (elementArray[parent] != root and elementArray[parent].symbol = '+')
{
    // the following get_sibling function retrieves the position of the
    // sibling node
    int sibling = get_sibling(elementArray[x], parent);
    // the following get_side_of_sibling function retrieves the branch
    // of the sibling node this will return either 'right' or 'left'
    char side = get_side_of_sibling (elementArray[x], parent);
    // get all the phis in the sibling node
    siblings_children_phis[] = get_all_phis(elementArray[sibling]);
    // get all the alphas in the sibling node
    siblings_children_alphas[] = get_all_alphas(elementArray[sibling]);
    // if the sibling is on the right side of the parent node
    if side = 'left'
        // and also if the sibling node is a phi. the element_is_a_phi
        // function returns true if the passed element is an alpha
        if element_is_a_alpha(elementArray[sibling])
            indirect_impact.push (elementArray[sibling].symbol);
        else
            no_cautionary_impact.push(siblings_children_phis[],siblings_children_alphas[]);
    }
    else {
        //element_is_a_alpha function returns true if the passed element is
        //a alpha
        case element_is_a_alpha(elementArray[sibling].symbol):
            no_cautionary_impact.push(elementArray[sibling].symbol);
        case element_is_a_phi(elementArray[sibling].symbol):
            indirect_impact.push(elementArray[sibling].symbol);
        case elementArray[sibling].symbol = '.':
            secondary_impact.push(siblings_children_phis[]);
            indirect_impact.push(siblings_children_alphas[]);
        case elementArray[sibling].symbol = '+':
            secondary_impact.push(siblings_children_phis[]);
            indirect_impact.push(siblings_children_alphas[]);
        }
    parent = get_parent(elementArray[parent]);
}
This is the detailed pseudo code associated with the implementation given in Appendix A. The comments given within this pseudo code refers to the Table 8.1 in Chapter 8.

//This step initially identifies the element that is changing. For this purpose, standard tree traversal methods are used.
search the KAT expression binary tree and locate the impacted_element

//Based on the element that is changing, the parent node and the current position is identified
parent_node = parent of the impacted_element
current_position = impacted_element

//Now this loop traverse the tree until it reaches the root element until parent_node != root_element of the binary_tree
{
  //As it traverse upwards in each step, the sibling node and children of siblings are identified based on the current position
  sibling_node = sibling of the current_position
  get all the children_of_sibling

  //Now we analyse whether the changing element is a phi
  //if the changing element is a phi it will execute this if the impacted_element is a phi
  {
    //this looks the positioning of the changing element whether it is on left or right
    //Only if it is on left the following will be executed if the impacted_element is on the LEFT branch of the parent node
    {
      //This case statement looks at all possible combinations that could exist for the //parent node and the sibling nodes
      CASE: parent_node is a dot (.) AND sibling_node is a phi
      {
        mark sibling_node as a secondary_impact
      }
      CASE: parent_node is a dot (.) AND sibling_node is a dot (.)
      {
        mark children_of_sibling as non_cautionary_impact
      }
      CASE: parent_node is a dot (.) AND sibling_node is a plus (+)
      {
        mark children_of_sibling as non_cautionary_impact
      }
      CASE: parent_node is a dot (.) AND sibling_node is a star (*)
      {
        mark children_of_sibling as non_cautionary_impact
      }
      CASE: parent_node is a dot (.) AND sibling_node is a negate (~)
      {
        if children_of_sibling are phi
        {
          mark children_of_sibling as non_cautionary_impact
        }
      }
      CASE: parent_node is a plus (+) AND sibling_node is a phi

\{ 
mark the sibling_node as a non_cautionary_impact 
\} 

CASE: parent_node is a plus (+) AND sibling_node is a dot (.)
\{ 
if children_of_sibling are phi 
\{ 
mark children_of_sibling as non_cautionary_impact 
\} 
\} 

CASE: parent_node is a plus (+) AND sibling_node is a plus (+)
\{ 
if children_of_sibling are phi 
\{ 
mark children_of_sibling as non_cautionary_impact 
\} 
\} 

CASE: parent_node is a plus (+) AND sibling_node is a negate (~)
\{ 
if children_of_sibling are phi 
\{ 
mark children_of_sibling as non_cautionary_impact 
\} 
\} 

//This part will be only executed if the changing element is of the right hand side of the //parent node 
else if the impacted_element is on the RIGHT branch of the parent node 
\{ 
//This case statement looks at all possible combinations that could exist for the //parent node and the sibling nodes 
CASE: parent_node is a dot (.) AND sibling_node is an alpha 
\{ 
mark the sibling_node as an indirect_impact 
\} 

CASE: parent_node is a dot (.) AND sibling_node is a phi 
\{ 
mark the sibling_node as a secondary_impact 
\} 

CASE: parent_node is a dot (.) AND sibling_node is a dot (.) 
\{ 
if children_of_sibling are phi 
\{ 
mark children_of_sibling as secondary_impact 
\} 
else if the children_of_sibling are alphas 
\{ 
mark children_of_sibling as indirect_impact 
\} 
\} 

CASE: parent_node is a dot (.) AND sibling_node is a plus (+)
\{ 
if children_of_sibling are phi 
\{ 
mark children_of_sibling as non_cautionary_impact 
\} 
else if the children_of_sibling are alphas 
\{ 
mark children_of_sibling as indirect_impact 
\} 
\}
CASE: parent_node is a dot (.) AND sibling_node is a star (*)
{
    if children_of_sibling are phi
    {
        mark children_of_sibling as non_cautionary_impact
    }
    else if the children_of_sibling are alphas
    {
        mark children_of_sibling as secondary_impact
    }
}

CASE: parent_node is a dot (.) AND sibling_node is a negate (~)
{
    if children_of_sibling are phi
    {
        mark children_of_sibling as secondary_impact
    }
}

CASE: parent_node is a plus (+) AND sibling_node is an alpha
{
    mark sibling_node as a non_cautionary_impact
}

CASE: parent_node is a plus (+) AND sibling_node is a dot (.)
{
    if children_of_sibling are phi
    {
        mark children_of_sibling as non_cautionary_impact
    }
}

CASE: parent_node is a plus (+) AND sibling_node is a plus (+)
{
    if children_of_sibling are phi
    {
        mark children_of_sibling as non_cautionary_impact
    }
}

CASE: parent_node is a plus (+) AND sibling_node is a negate (~)
{
    if children_of_sibling are phi
    {
        mark children_of_sibling as non_cautionary_impact
    }
}

//This part is executed if the changing element is an alpha
else if the impacted_element is an alpha
{
    //f the changing element is on the left hand side of the parent
    node, this part is executed
    if the impacted_element is on the LEFT branch of the parent node
    {
        //This case statement looks at all possible combinations that could
        exist for the //parent node and the sibling nodes
        CASE: parent_node is a dot (.) AND sibling_node is an alpha
        {
            mark sibling_node as an indirect_impact
        }
    }
}
mark children_of_sibling as non_cautionary_impact
}
CASE: parent_node is a dot (.) AND sibling_node is a plus (+)
{
    mark children_of_sibling as non_cautionary_impact
}
CASE: parent_node is a dot (.) AND sibling_node is a start (*)
{
    if children_of_sibling are alphas
        mark children_of_sibling as non_cautionary_impact
    }
}
CASE: parent_node is a dot (.) AND sibling_node is a negate (~)
{
    if children_of_sibling are phis
        mark children_of_sibling as non_cautionary_impact
    }
}
CASE: parent_node is a plus (+) AND sibling_node is an alpha
{
    mark sibling_node as a non_cautionary_impact
}
CASE: parent_node is a plus (+) AND sibling_node is a dot (.)
{
    mark children_of_sibling as non_cautionary_impact
}
CASE: parent_node is a plus (+) AND sibling_node is a plus (+)
{
    mark children_of_sibling as non_cautionary_impact
}
CASE: parent_node is a plus (+) AND sibling_node is a star (*)
{
    if children_of_sibling are alphas
        mark children_of_sibling as non_cautionary_impact
    }
}

//This part is executed only if the changing element is an alpha and
//is on the right hand side of the //parent node
else if the impacted_element is on the RIHGT branch of the parent
{
    //This case statement looks at all possible combinations that could
    exist for the //parent node and the sibling nodes
    CASE: parent_node is a dot (.) AND sibling_node is an alpha
        mark sibling_node as a non_cautionary_impact
    }
    CASE: parent_node is a dot (.) AND sibling_node is a phi
        mark sibling_node as an indirect_impact
    }
    CASE: parent_node is a dot (.) AND sibling_node is a dot (.)
        if children_of_sibling are phis
        mark children_of_sibling as secondary_impact
    }
else if children_of_sibling are alphas
{
mark children_of_sibling as indirect_impact
}
}
CASE: parent_node is a dot (.) AND sibling_node is a plus (+)
{
if children_of_sibling are phis
{
mark children_of_sibling as secondary_impact
}
else if children_of_sibling are alphas
{
mark children_of_sibling as indirect_impact
}
}
CASE: parent_node is a plus (+) AND sibling_node is an alpha
{
mark sibling_node as a non_cautionary_impact
}
CASE: parent_node is a plus (+) AND sibling_node is a dot (.)
{
mark children_of_sibling as non_cautionary_impact
}
CASE: parent_node is a plus (+) AND sibling_node is a plus (+)
{
mark children_of_sibling as non_cautionary_impact
}
CASE: parent_node is a plus (+) AND sibling_node is a star (*)
{
if children_of_sibling are alphas
{
mark children_of_sibling as non_cautionary_impact
}
}
//it collects new values for the current position and parent node as
//it traverse up the tree

current_position = parent_node
parent_node = parent of the current_position
}