Intelligent Monitoring of Business Processes using Case-based Reasoning

PhD

Stylianos Kapetanakis

School of Computing and Mathematical Sciences
University of Greenwich
London, England

A thesis submitted in partial fulfilment of the requirements of the University of Greenwich for the Degree of Doctor of Philosophy

June 2012

Declaration

I certify that this work has not been accepted in substance for any degree, and is not concurrently being submitted for any degree other than that of Doctor of Philosophy (Ph.D.) being studied at the University of Greenwich. I also declare that this work is the result of my own investigations except where otherwise identified by references and that I have not plagiarised the work of others.

Stylianos Kapetanakis

Supervisors

Professor Miltos Petridis

Professor Liz Bacon

Professor Kevin Parrott

Acknowledgements

I would like to express my gratitude to Professor Miltos Petridis for his unrivalled guidance, the support and the inspiration that he gave me over the past years. He was a real family to me and this thesis is here thanks to him.

I would also like to thank Professor Brian Knight for his suggestions and Professor Liz Bacon for her comments during the PhD enactment. I am grateful to Roger Cuthbert and his company for their kind data provision.

Thanks to Doctor Nikos Papadakis and Doctor Nikos Christakis, my colleagues and friends that guided me all over my journey five years ago. Thanks also to Doctor Chris Koulopoulos for his invaluable help and support.

I would like to thank Doctor George Samakovitis for his encouragement during the difficult period of writing-up.

Thanks to John Agorgianitis for being a real friend at all times. Thanks to a very special person standing by me all the sleepless nights till early in the morning while writing this thesis.

Thanks also to the University of Greenwich for the funding, my fellow colleagues and the excellent technical support.

Finally, I would like to thank my family and friends, all the hidden heroes in various aspects throughout this thesis as well as everybody else that was not mentioned above.

Dedicated:

to Stef for his unrivalled support in life.

Abstract

.

The work in this thesis presents an approach towards the effective monitoring of business processes using Case-Based Reasoning (CBR). The rationale behind this research was that business processes constitute a fundamental concept of the modern world and there is a constantly emerging need for their efficient control. They can be efficiently represented but not necessarily monitored and diagnosed effectively via an appropriate platform.

Motivated by the above observation this research pursued to which extent there can be efficient monitoring, diagnosis and explanation of the workflows. Workflows and their effective representation in terms of CBR were investigated as well as how similarity measures among them could be established appropriately. The monitoring results and their following explanation to users were questioned as well as which should be an appropriate software architecture to allow monitoring of workflow executions.

Throughout the progress of this research, several sets of experiments have been conducted using existing enterprise systems which are coordinated via a predefined workflow business process. Past data produced over several years have been used for the needs of the conducted experiments. Based on those the necessary knowledge repositories were built and used afterwards in order to evaluate the suggesting approach towards the effective monitoring and diagnosis of business processes.

The produced results show to which extent a business process can be monitored and diagnosed effectively. The results also provide hints on possible changes that would maximize the accuracy of the actual monitoring, diagnosis and explanation. Moreover the presented

approach can be generalised and expanded further to enterprise systems that have as common characteristics a possible workflow representation and the presence of uncertainty.

Further work motivated by this thesis could investigate how the knowledge acquisition can be transferred over workflow systems and be of benefit to large-scale multidimensional enterprises. Additionally the temporal uncertainty could be investigated further, in an attempt to address it while reasoning. Finally the provenance of cases and their solutions could be explored further, identifying correlations with the process of reasoning.

Table of Contents

List of Tables

List of Figures

Abbreviations and Acronyms

AI	Artificial Intelligence
AHC	Agglomerative Hierarchical Clustering
ANN	Artificial Neural Network
API	Application Programming Interface
BPM	Business Process Management
BPMI	Business Process Management Initiative
BPMN	Business Process Management Notation
BTS	Box Tracking System
CBR	Case-based Reasoning
CBR-WIMS	Case-based Reasoning Workflows Intelligent Monitoring System
CCC	Computer Cooking Competition
CMS	Computing and Mathematical Sciences
CRM	Customer Relationship Management
DHC	Divisive Hierarchical Clustering
DSS	Decision Support System
EMS	Exam Moderation System
ERP	Enterprise Resource Planning
GDP	Gross Domestic Product
GE	General Electric
GPS	Global Positioning System
HC	Hierarchical Clustering
HRM	Human Resource Management
HSDPA	High-speed Downlink Packet Access
ICCBR	International Conference in Case-Based Reasoning
IT	Information Technology
kNN	k-Nearest Neighbour
MBR	Model Based Reasoning
MCS	Maximum Common Sub-graph
OASIS	Organization for the Advancement of Structured Information Standards
OMIS	Organizational Memory Information Systems

OMG	Object Management Group
ORB	Object Request Broker
RBES	Rule-based Expert System
RBS	Rule-based System
RDF	Resource Description Framework
SCA	Service Component Architecture
SOA	Service-oriented Architecture
TCBR	Temporal Case-Based Reasoning
UML	Unified Model Language
WFMS	Workflow Management System
Wi-Fi	Wireless Fidelity (refer to: WLAN)
WLAN	Wireless Local Area Network
WS-BPEL	Web Services Business Process Execution Language
WSDL	Web Services Definition Language
XPDL	XML Processes Definition Language
YAWL	Yet Another Workflow Language

Chapter 1

Introduction

“The most incomprehensible thing about the world is that it is comprehensible.”

Albert Einstein

Modern organisations employ a large number of business processes and procedures to ensure the accomplishment of their goals. The effective management, monitoring and troubleshooting of these processes is of paramount importance to the organisation.

With the help of business processes large, complex organisations can precisely set their goals, refine their procedure mechanisms and manage their multi-dimensional resources regardless of their actual diversity, coherence and internal complexity. However, this complexity in combination with the accumulative communication load among any internal/external stakeholders can cause a significant overhead to the business processes while increasing in size and range.

Business processes for more than twenty years [Davenport & Short, 1990] have been seen as tightly connected with information and communication technologies which are used extensively in current world economies. Latest advances have shown that their application in communities and organisations enhances innovation dramatically, increases productivity and affects the current standards globally. The

10th Global Information Technology Report [World Economic Forum, 2011] highlights how their application is affecting 138 economies globally, constituting over 98 per cent of the global gross domestic product (GDB).

Business processes can be relatively complex, thus their monitoring is usually a human task. However, nowadays their monitoring requires automation, due to the high complexity of information within processes and the large volume of data involved. A challenge associated with the monitoring of business processes is whether they can be automated by using specialised software.

This work has been triggered by the above challenge. It investigates whether there can be automated monitoring as well as up to which extent this monitoring can be applied. This chapter explains the rationale behind this work as well as the formulated research questions. Finally a summary of this thesis is presented.

1.1 Rationale

The term business process is relatively new. However, the implied concept has existed for a long time. Business processes can be regarded as procedures that contain several steps/stages that have to be satisfied towards a pre-defined goal, result or product.

Business processes nowadays are being controlled and/or monitored via enterprise computer systems where their orchestration is being controlled by software systems. The monitoring of any complex situation is usually a non-automated task that resides within the responsibilities of human managers. These managers have to deal with any emerging, complex or unanticipated situations.

Traditionally business process managers deal with process monitoring in a non-automated way. However, nowadays the number of processes involved in a typical business function is quite high and complex. The volume of data that has to be considered, evaluated and be subject to monitoring is equally high. Given the above, a challenge and an opportunity are being created: firstly to investigate whether there can be sufficient and effective monitoring to business processes and secondly, if this is the case, up to which extent this monitoring can be automated.

Since Business processes are of such importance to the modern world, their definition from a technological aspect should be defined formally. Current standards, such as UML, BPMN and WS-BPEL [IBM, 2007], provide the essential formalism for effective and efficient Business Process representation across industries, ensuring consistency among procedures. An efficient representation of a business process can promote its consistency, improve communication in terms of the definition of operations (e.g. who is supposed to do what) as well as afford easier (re)design/ (re)implementation of its consisting parts.

A typical example that can illustrate the above is a loan approval procedure in the banking sector. A loan approval is a business process that consists of a number of sub-processes. These can be seen in figure 1-1 and can include:

- an initial application, containing the borrowing needs of the client, information regarding his/her income, liabilities, assets, previous loans, etc.
- the underwriting process where supporting documents are provided along with the credit history of the customer
- the processing stage which requires contact with the borrowers, the realtor, the appraiser, the loan officer, the credit report company, the title company and the escrow company [East West Bank, 2012]
- the loan closing document and funding stage where the final documentation follows and the loan can be characterised as finished

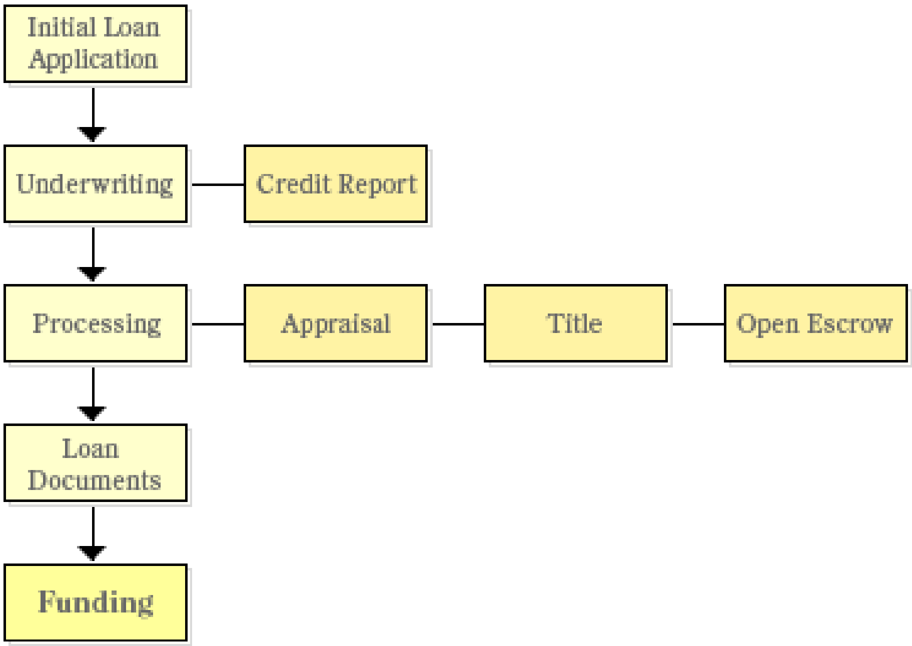

Figure 1-1: Loan approval business process [East West Bank, 2012].

As seen in figure 1-1 the loan process involves a significant number of systems that may reside in the same company or be outsourced to others (e.g. the credit approval). The workflow of the above process can be captured into some formal or diagrammatic representation but its monitoring should be conducted via an expert or a team of experts. This has to be done manually in order to deal with an unanticipated situation or problem. An efficient representation allows the modular approach of the process design, allowing parts to be changed on demand (e.g. allocate the credit checks to a different credit rate agency, add a different part to the Processing part, etc.)

Since business processes are of such wide use, their efficient management is important. Modern business process management (BPM) covers business processes across organisational boundaries, coordinating the flow of information among vendors and allowing adaptation to changing conditions in a synergistic manner [Liu, Li & Zhao, 2009]. However, in reality this constitutes a perennial problem which states a hard to avoid business challenge. Efficient management means that in any current enterprise, that encapsulates business processes, its "operational efficiency is maintained at sufficiently high levels, such that the return on investment is sustainable enough to justify its continued existence" [Bandara et al., 2007].

Current business process technologies, although functional enough in terms of modelling, are failing to meet challenges in terms of governance, organisational strategy, process design and finally their actual execution. Therefore there is a challenge to enhance the existing amalgamation of technologies with a top level intelligent management of their underlying processes. Moving even further from the design, representation and execution of a business process, the challenge is whether we can effectively monitor, analyse in depth and remove the fuzziness associated with the complexity of business execution. Another challenge is to be able to provide feedback to the investigated system based on the knowledge that a monitoring system has collected. A key challenge is also how to coordinate the existing technologies and the information collected by a process monitoring system to provide feedback and effective management in a practical and effective way.

In most processes nowadays a common phenomenon is being observed: while reaching a stage of automation, the volume of any acquired data is significantly higher compared to the volume that can be controlled by human managers and existing tools [Rozenfeld, 2007]. A descriptive example relating to the above is the control of current camera surveillance systems worldwide. From their initial application, their control was considered relatively trivial, having allocated a certain number of staff over a limited number of devices. However, the numbers have significantly climbed up in recent years both in terms of operating devices and data generated, making their efficient monitoring significantly harder. London metropolitan police announced some years ago that it cannot deal with the volume of information generated by its surveillance systems [CNET, 2009]. The control of such systems can be regarded as a really intensive task and a way to automate it could be of significant benefit. Of course the question raised at this point is to which extent this monitoring can be automated. Considering the business processes and the challenges associated with them, this research is investigating the feasibility of applying intelligent monitoring based on their unique characteristics.

Business process managers dealing with problematic situations appear to work mainly by remembering previous sequences of events[1]. These events are associated with past problematic executions which have been subject to remedial actions and/or some reflection. The managers are reusing this knowledge, gained by previously applied solutions, taking into consideration the characteristics of the investigated process. This is very similar to the Case-based Reasoning paradigm. Case-based Reasoning (CBR) [Kolodner, 1993] is an Artificial Intelligence technique that operates in a similar way, reusing knowledge and adapting it to the problem in hand. It is therefore worth investigating its application within the concept of business process monitoring.

1.2 Main Research Question

The main research question that this thesis is called to answer is the following:

Can Business Processes be intelligently monitored and diagnosed with the use of software techniques?

More specifically, **can techniques based on Case-based Reasoning (CBR) assist effectively in the intelligent monitoring and diagnosis of business processes?**

CBR systems offer an effective approach to real-time problems in a number of cases [Kolodner, 1993]. CBR techniques have been used across different environments in order to support the decision-making

[1] Actions and events throughout this thesis refer to the same entity.

process for human correspondents. The choice of CBR is due to its main characteristic to mimic the way that human experts apply their experience to problem solving by remembering similar past experiential cases. This constitutes a natural approach to problem solving making it the most obvious way of pursuing this research. Other artificial intelligence methods and techniques such as Artificial Neural Networks (ANNs) or Rule-based Systems (RBSs) can be used but they carry a number of disadvantages that make their application to this problem challenging. A detailed discussion regarding these will follow in Chapter 2.

This thesis investigates whether CBR techniques can be used to address the problem of monitoring and diagnosis of a Business Process that is being executed. Monitoring is a "real time" or a simulated "real time" investigation of the current state of a workflow. Diagnosis includes the monitoring along with the provision of the related context and explanation of the current workflow state. Additionally the state associated knowledge or advice is provided that may be useful to a workflow stakeholder.

Since business processes are being heavily monitored by human managers, the need for effective recommendations is preferred to already-made decisions. A system that is delegated to support humans should be transparent through its operation, indicating the rationale (what, how, why) behind any stated assumption/recommendation. This can be part of the explanation provided in order to build confidence with its users. CBR systems are transparent enough and can supply explanation while presenting their output [Roth-Berghofer, 2004; Sørmo et al., 2005; Kapetanakis et al., 2010b]. This adds clarity to the decisions taken and progressively builds confidence between the system and its users.

This research makes an in depth investigation on whether CBR can be adopted as a primary means towards the effective intelligent monitoring and diagnosis of Business processes.

1.3 Subsidiary Questions

The main research investigation triggers several subsidiary questions, which are briefly described below.

What is a suitable representation of a business process execution to allow the effective application of the CBR process?

While attempting to monitor a business process the need for computer representation of business process attributes and rules emerges. Business processes can be defined in a standardised way via BPMN, but during the monitoring of a running business process the representation of the definition is not directly used. Most of the existing standards like BPEL have been developed to clarify the insights of a business process to humans as well as to represent the choreography of actors working within the system.

Existing business process systems follow instructions orchestrating the running of systems services based on human-designed business processes descriptions. However, such systems do not have mechanisms to monitor these processes. A CBR approach to the intelligent monitoring of business processes requires a mechanism to measure similarity between real execution instances of business processes. These process execution instances will be referred to as traces for the rest of this thesis.

In order to reuse knowledge from similar workflows and define the similarity among them a suitable formal representation is required. A formal workflow representation is needed in order to achieve its efficient monitoring. Current business modelling standards offer an accurate representation of the business processes. However they lack in terms of formalism since there can be multiple different representations for the same process. A suitable representation should follow a mathematical formalism and allow only one representation for a business process execution.

What are effective similarity measures between workflows?

A system that monitors a business process should be able to reuse knowledge from similar past cases. Past cases from the knowledge repository should be associated with a monitored case. In order for this to happen an appropriate similarity measure should be applied among the business process instances. A similarity measure should be defined but the structural complexity of workflow cases makes this a challenge. This is especially the case in systems where temporal aspects and uncertainty are involved. In such systems the successful establishment of suitable similarity measures is even more challenging.

How can the monitoring process and results be explained to users of the system?

In order for the monitoring to be effective and due to the fact that similarity measures are not always clear to humans, it's important to provide an insight into the similarity between cases. Humans have difficulty understanding the similarity between compared temporal sequences. Therefore visualisation of the similarity measures can be used to extract reflective insights that combine and explain the process context, the associated knowledge and the suggested prospective actions.

The system should be able to provide the necessary information in order to convince the human operators why something is the same with something else, how was that measured and what was the evidence from the past that leads towards the suggested action. In a system that monitors business processes it is also worth investigating the provenance of a particular case along with the way its neighbours were retrieved and suggested.

What software architecture will support a CBR system to effectively monitor business process execution?

Usually the orchestration and execution of a business process involves complex, enterprise-wide systems, which can be more than one in number. In order for an intelligent monitoring system to work efficiently, it has to be compatible with current standards. Additionally it should provide an architecture that allows it to be deployed easily within an existing enterprise system. Most workflow systems are real time systems so an efficient monitoring system should have close integration to operations and should not affect their integrity and operational efficiency. Workflows and systems that control workflows are subject to continuous changes, something that makes the definition of a suitable architecture a necessity. Finally as knowledge is to be used across systems, it is important to have an architecture that can work for more than a specialised system and/or a specialised format. Knowledge acquired from the operation of a system over a number of years should be transferable between different versions of the system and to and from other systems that work in a similar way. An example of that could be borrowed from the banking sector. Ideally the knowledge stored in a system that works for years on loan approvals contains knowledge that could be useful for a system working on the approval of mortgages after being subject to the necessary adaptations.

1.4 Thesis Outline

Chapter 2 provides the necessary theoretical background that current research has been based on in terms of business process standards, workflow representation and visualisation, log mining, graphs, graph similarity, similarity algorithms and the general time theory. Chapter 2 also refers to the existing work on CBR systems as well as its application in the workflow domain. The motivation plus the discussion of the approach adopted in this thesis is also presented.

Chapter 3 contains discussion regarding the motivation behind the conducted experiments, as well as the followed approach in terms of the methodology of this research. The algorithms used for similarity measures are presented as well as indicators that lead to the derived architecture of the proposed system.

Chapter 4 discusses the findings in terms of the system validation using a wide range of datasets. Their range varies from simplified to more complex ones. The rationale and conclusions from these experiments are also presented covering a number of aspects.

Chapter 5 provides an extensive description and evaluation of the architecture which was implemented for the needs of intelligent business process monitoring. For the needs of this research the framework was deployed on a second enterprise workflow system to evaluate the suitability of the architecture. The extended evaluation of the architecture on the new case study is presented as well as enhancements to the architecture that have been developed and tested to ensure the portability and transparency of the proposed system across a wider range of business process monitoring situations.

Chapter 6 discusses how the concept of explanation has been adapted to enhance the effectiveness of intelligent monitoring of business processes. The provision of explanation and provenance in the proposed system is shown and evaluated using both case study systems. The related research work is discussed here along with the related experiments and conclusions.

Chapter 7 contains the conclusions of the thesis as well as the results found throughout this research investigation. The key research contributions are presented and appraised. Discussion on how this research work can be extended and future plans for further work are also included.

Chapter 2

Motivation: Intelligent Management of Business Processes

2.1 Introduction

Business processes define a relative wide spectrum of the modern industrial world. Business processes can define precisely the required procedures within an organisation, ensuring aspects such as the industrial standards, the quality and faultless operation of a production chain, delivery of services, etc. The flow of a business process can be defined by a team of qualified organisation experts. Their role is to pin down with accuracy how tasks, actors and resources have to be used in order to achieve a pre-specified goal.

Current standards and representations of business processes give an insight into their definition. Workflows can show how the processes are being executed and modified over time. The existing technology can give answers to questions like what, where or when something occurred within the execution of a business process. However, adequate answers cannot be provided to questions like: why this incident took place in such way?, or what should be done in this unanticipated situation?, in other words to understand and manage the business process in an intelligent way.

Workflows have been proven an effective way towards the orchestration and choreography of a business process [Jung et al., 2004]. Therefore their effective monitoring is needed for the establishment of an intelligent business process management. To achieve that Artificial Intelligence (AI) could be used, to assist the workflow monitoring by using any available past experience. Case-based Reasoning could be identified as a potential technique towards the above goal. This chapter gives an overview of existing CBR systems that have worked effectively in the areas of event mining, agile workflows and other related interdisciplinary fields.

When a Business process is being executed it can generate a significant volume of data that can be stored in a variety of formats via physical means. The data produced is usually related to the executed actions in either a loose (informative) or a tight (critical) way. Based on the existing data an authorised manager can recreate the executed path of the business process. This path can afterwards reveal information related to what happened in an investigated past case.

In order to follow the execution of a business process the need for its precise representation is being formulated. Towards its actual representation a workflow could be used since its internal structure can be efficient. A workflow can be formulated by sequences of events or tasks which are being triggered by specific actor roles, already defined in the system. Such an approach seems expedient towards the precise representation of a business process execution.

While a business process is being executed a number of actions can take place. These actions may be related to different processes and could have a certain sequence. Actions could be broken down into smaller pieces where each of them can constitute a basic atomic event element in the structure of the business process. When a business process is being executed a lot of such events are being generated and captured.

The executed snapshots of a business process are being based on events and their related temporal information. Towards the effective interpretation of such events and their temporal relations it seems fundamental to be able to understand their followed sequence.

Understanding of their temporal flow seems vital as well. This flow could be possibly represented in terms of a time theory which can identify the reference of specific events in time with indicators like what happened first, what next, etc. The value of the time theory is being stressed when there is a need to extract information from the consecutive executed sessions of a business process. In such cases a formalised temporal approach is needed in order to be able to understand and measure the similarity of the different time traces.

The mining of the events and their representation in a workflow seems an obvious approach towards the monitoring of a business process. However, the sequences of events may not contain a meaningful context if they are not presented in an appropriate way. The same applies when attempting to estimate the similarity among different sequences, since the sequences by themselves cannot be compared. In order for this to happen, the event sequences could be represented in terms of a graph and then the overall problem pertains to the area of graph similarity. Graph theory could be of use in such case.

Business process workflow instances could be possibly represented by graphs. Based on this representation Case-based Reasoning (CBR) techniques could be applied afterwards, in order to extract useful patterns that could assist human actors in decisions making.

The work in this thesis deals with the wide concept of Business Processes and investigates whether they can be subject to efficient Monitoring and Diagnosis. In order to be able to unfold smoothly the conducted work, some fundamental entities regarding Business Processes have to be defined in advance. Since the work overall operates within the context of Business processes, their existing standards will be initially presented in this chapter.

This chapter investigates how workflows relate to the business processes in terms of their effective representation and their related technologies that can be used for their effective monitoring. The execution of a business process can lead to a production of a large volume of events and their relevant temporal information. Therefore the current state in the time representation is being investigated.

Since monitoring involves temporal data the crucial factor of Uncertainty in Temporal Data is also discussed in an attempt to provide a solid workbench for the rest of the thesis. The purpose of the current chapter is to present the work conducted in the above fields and present their internal association on which the current research has been based.

The research investigation in this thesis is highly related with the area of Artificial Intelligence using Case-based Reasoning (CBR). This chapter presents an overview of the area as well as the related work in CBR with Business Processes. Finally CBR work conducted in the area of Workflow adaptation is presented in addition to the work on CBR applications on Workflows and Business Processes respectively.

The rest of the chapter is organised as follows: Section 2.2 discusses business processes and their existing representation standards. Section 2.3 relates to the representation of business processes as workflows, the concept of workflow control and its intelligent management. Section 2.4 analyses how sequences of events could be represented in terms of a temporal time theory as well as which other theories exist in the area. Section 2.5 presents an overview of the CBR and its various application aspects in general systems, event-mining and the workflows. The concepts of uncertainty, adaptation and the temporal role in CBR are also discussed in the same section. Finally section 2.6 presents the conclusions of this chapter along with a brief summary of its visited concepts.

2.2 Business Processes & Standards

Business processes can be found in a wide range of applications. Their operational spectrum is significantly wide including legal organisations (such as governments or non-governmental organisations), corporations (either of profit or not-for-profit ones), international organisations, universities, partnerships and even companies of a rather small, medium, large and very large size.

It could be questioned whether business processes are tightly connected with processes, what is their actual interrelation or whether they have any connection overall. An early definition that could possibly help goes back to 1776. Adam Smith at his book Wealth of Nations first referred to what could be characterised as an ancestor of a process. That was actually one of the first definitions for an industrial process and is quoted below. At this point is worth mentioning that Smith gave the process definition in order to be able to explain a product chain process: the pin factory product chain. "One man draws out the wire, another straights it, a third cuts it, a fourth points it, a fifth grinds it at the top for receiving the head; to make the head requires two or three distinct operations; to put it on is a peculiar business, to whiten the pins is another; it is even a trade by itself to put them into the paper; and the important business of making a pin is, in this manner, divided into about eighteen distinct operations, which, in some manufactories, are all performed by distinct hands, though in others the same man will sometimes perform two or three of them." [Smith, 1910].The above definition gives relevance on how the process was identified even from a long time ago in terms of an industrial perspective.

Business processes are related with the industrial processes, preserving a different perspective compared with the above illustrative example. A Business process could be defined as a “structured, measured set of activities designed to produce a specific output for a particular customer or market" [Davenport, 1993]. A possibly more simplistic definition can be found from Hammer & Champy (1993) where a business process can be referred to as a "collection of activities that takes one or more kinds of input and creates an output that is of value to the customer". Using the word customer in a definition for business processes could be inappropriate since the meaning of customer is rather focused and indicates sales, something that may mislead the reader. Business processes are widely usable and not tightly connected to the term of customer. As a result a more direct and generalised definition can be formulated. A Business Process could be regarded as "a set of practices performed to achieve a given purpose and satisfy a *business need"* [Phillips, 2004].

Business processes can be defined to serve a specific purpose that has to be run once. Alternatively they could illustrate a business pattern that could be repeated eternally. Organisations have identified early the need for business process standardisation since significant amounts of their resources and budget were spent on the fundamental stage of design. In many cases no matter how good the initial design was it had to be followed by a redesign of either the whole business process or parts of it. The main reason for that was due to the lack of standards in design which led to unavoidable misunderstandings and consequently to a whole series of glitches.

Generally non-standard designs lead to a potential problematic system implementation. That successively can lead to insufficient maintenance of the on-going arrangements. The operational landscape can become even worse if a change has to be applied to an existing business model. A working business model which is subject to constant changes can have multiple interactions among actors. The large numbers of actors and interactions lead to higher levels of complexity. This complexity seldom allows the smooth operation of a model; it is more possible to lead to successive faults. An eternal fault sequence unavoidably leads to a major redesign or overall restoration of the process.

The standardisation of a business process offers the required guarantee for the development of large systems. Based on that there is no issue on where, when or how the various components are implemented, since the final product will comply with a set of generally accepted rules. These broadly accepted rules fortify the product, ensure its integrity and can easily integrate it into new operational components.

Careful design based on a widely acceptable rule-set and its correspondent procedures, allows collaboration among different vendors regardless of their software engineering tools and methodologies. Moreover specialised vendors can get involved within any development and/or design phase of a system. Standards offer the necessary sanctuary for clear system definition and wide familiarisation. The enforcement of standards also helps systems to escape the curse of being black boxes and as a result one-use boxes.

A set of standards for a business process should cover the “definition, orchestration and choreography of business processes” [Kapetanakis et al., 2009a]. The need for standards is triggered from the need of a widely acceptable formalism in the representation of business processes. Formal standards have been developed towards a more specific, clear and widely acceptable business process. Their presence eradicates up to a certain extent the hassle of different business process semantics, different proprietary formats and dissimilarities in run time representations [IBM, 2007]. Business processes attain the necessary interoperability via these globally accepted rules and definitions and can be used afterwards with relative ease across different industrial vendors. Reasonably this can work effectively towards their benefit.

2.2.1 Business Process Representation

Business Process Standards can ensure that once a business process is defined in one of their universally acceptable notations, it can be transferred to a different system without information loss. An imposed standard can also ensure that additions, removals and alterations of the initial prototype will run in the same way across systems that support the addressed standard.

Towards the broad standardisation of Business Processes, OASIS (Organization for the Advancement of Structured Information Standards) has proposed the WS-BPEL [OASIS, 2007a] as "an execution language to describe the behaviour of business processes in a standards-based environment" [IBM, 2007]. WS-BPEL or BPEL in short, describes the orchestration among business processes and external actors via web services.

BPEL enables the execution of web transactions in a wide environment of business processes. Transactions can be either within the processes or among the processes. The processes themselves can be either Executable or Abstract [IBM, 2003]. Executable processes are meant to support behavioural characteristics or interaction among business process actors whereas Abstract ones are meant to support the business process descriptive characteristics. Abstract processes may never come in action and this could be a reason why an abstract business process intentionally keeps its operational information hidden.

The Web Services used in BPEL can be formulated in terms of the Web Services Definition Language (WSDL) [W3C, 2001]. Although Web Services face limitations such as the lack of state persistence or message cohesion, BPEL manages to overcome these limitations by assigning the important roles to the persistent parts of the infrastructure. These parts are the two edges of the established communication: the actual business processes.

Since the BPEL standard does not deal with all the needs of representation in the world of Business Processes, several proposed extensions have been derived. Most of them are based on BPEL in order to meet the market needs. BPEL4People [IBM & SAP, 2005] is an extension to BPEL that covers the aspect of how people specifically interact and generally contribute to a business process while it's being executed. BPEL4People refers to scenarios [IBM & SAP, 2005] where the focus is on the people involved while everything is built on top of the native BPEL language.

Other extensions like BPELJ and WS-Business Activity deal with current limitations of the language and assist the Business Process engineers. BPELJ [IBM & BEA Systems, 2004] takes advantage of the power of Java programming language and BPEL at the same time. WS-Business Activity [OASIS, 2007b], an OASIS recognised standard, provides a coordination framework of how the involved long-running activities in a Business Process should collaborate.

The Business Process Management Initiative (BPMI) and Object Management Group (OMG) provide another standard for the effective representation of a Business Process called Business Process Modelling Notation (BPMN) [OMG, 2011]. BPMN allows Business Process Architects to define a Business Process to a high level and be able to drill-down to any sub level depending on the outcome they want to accomplish. BPMN operates like a bridge for the two vital components of modern systems, the design level and the IT operational level. Via BPMN it is possible to explain the key aspects of a Business Process not only in terms of definition but also in terms of its choreography and orchestration. BPMN, where used, ensures the consistency across users who deal with the process in a variety of ways.

XML Processes Definition Language (XPDL) [WfMC, 2008] is a standard produced from the Workflow Management Coalition (WfMC). XPDL deals with the elementary issue of archiving BPMN diagrams. XPDL provides a secure way in the form of a file format where BPMN-defined business processes can be initially archived and subsequently transferred among vendors. This alleviates the stresses of possible process misinterpretations, misuses and malfunctions. Moreover it maintains cohesion among vendors with different responsibilities such as process modelling, analysis and execution. Nowadays the exchange of Business Processes can be characterised as simplified due to the efficiency of the XPDL standard.

As the literature shows the existing standards offer a precise way for business processes to be efficiently and indisputably defined, stored and transferred among vendors. The existence and enforcement of business process standards helps towards the eradication of the consequent design and redesign efforts of the business processes. They also eliminate the confusion for different approaches in the notation of the same business process. The target for standardising and homogenising a business process has been achieved.

Standardisation of a business process via widely acceptable formats and notations provides a secure way for depicting, editing and following pre-defined procedures. This can cover the primitive needs for the operations of business processes, their efficiency and following maintenance. However, nowadays there is a tremendous number of executed operations coordinated via business processes [World Economic Forum, 2011]. Although their structural needs and design topology are covered with the existing standards they seem inefficient in terms of monitoring. A reason for that is that the monitoring takes place on the already executed business process sessions, which the current standards are not designed to cover (they are representation standards). Therefore it is questionable whether they can be effective in terms of the upcoming and exponentially increasing monitoring needs.

Current observations regarding the above, clearly state that a business process can be represented precisely but not necessarily monitored with the existing standards. For the needs of the monitoring an appropriate set of technologies and techniques has to be used. This set of imported technologies could probably be liaised with workflows since they can offer an in depth approach while the business process is being executed. Section 2.3 will investigate the potential that workflows offer towards that direction.

2.3 Business Processes as Workflows

As shown in section 2.2.1 Business Processes can be represented efficiently in a globally acceptable format that can ensure data integrity across platforms. An imposed challenge after having a well-represented business process, is to understand what is its actual outcome while being executed. An indicative example of that is whether a human expert is able to answer the questions "Who? What? When? In a business process" [Virdell, 2003], when the latter is being executed.

The answer to question Who should be given by being able to indicate clearly the actors and the interfered components in a business process. This relates to the involved human roles, available groups, provided services, etc. Answers related to the above question should also be able to reveal the hierarchy among the components of the business process.

The answer to What question should relate to what actions can be performed by the business process participants. An answer should also list all available operations and transactions as well as the level of authorisation that these actors have regarding the operations. A sub-question inside What could also be on whether the available types of actions are automated, manual or a combination of those.

Finally, the answer to When question should indicate:

- when does a process start
- when does it finish

- how could the indicated participants know when to perform
- what is the duration of a process
- when can a task be characterised as completed, pending, failed, etc.

Workflows could be suggested as an effective approach to the above challenges. Their default structure allows the separate definition of the business process (how) from the implementation of the actual enterprise system. This division allows possible workflow adaptation [Reichert & Dadam, 1998] when new business needs occur. In such occasions the workflows need to adapt in order to be able to meet the on-going, changing business process requirements.

A workflow representation seems convenient when dealing with business processes. Via the workflow an appropriate model can be built to describe the process, whereas at the same time all operational aspects can be inspected thoroughly. This can allow further process optimisation. The related actors, stakeholders and managers can be highlighted in this context in combination with their associated roles, tasks, dependencies and overall requirements.

Mentzas et al. (2001) did an investigation on how the workflow technology can possibly assist in the business process management. Managers could use workflows in an attempt to realise possible similarities of an investigated case compared with cases that happened in the past. This approach could identify differences in behaviour, possible abnormality and at least a general behaviour pattern for an inspected process. Towards this direction process-oriented workflow management systems (WFMSs) [Georgakopoulos et al., 1995; Leymann & Altenhuber, 1994] offer the necessary stability for the development of business applications since they depict the way in which a business process should be executed. Via Workflow management, Business Processes can have their workflows defined in terms of role allocation, task coordination and execution. This also offers the opportunity for (re) design and (re) implementation of their tasks to meet new needs and unanticipated changes.

A key advantage while using workflows is the significant freedom that exists in terms of the operational perspectives. Workflows can offer a large variety of leveraged perspectives inside an operational environment. This can be of significant advantage to the whole hierarchical ladder of involved actors in an organisational context.

2.3.1 Workflow Types

Workflows can be distinguished in several types based on the type of their operational context. These types according to Bates (1993) could be generalised in three categories: the Production workflow, the Administrative workflow and the Ad hoc workflow. These categories derive from the nature of the workflow application domain and they are briefly outlined below.

Production Workflows

The Production workflows relate to complex, heavy-structured applications, manufacturing product chains, bank process chains and product development life cycles [Bates, 1993]. A common pattern for these procedures is that they are task-driven. The people who are allocated to certain tasks can communicate with the next person in the product chain, seeking approval or continuing the flow of the process. Since Production workflows depict the current stage of the production cycle, they can anticipate heavy customer interaction. As a result they may need constant revision in order to meet the on-going changing requirements.

Administrative workflows

Workflows that deal with the forms of a specific task could be characterised as Administrative workflows [Bates, 1993]. These are usually simple tasks that contain certain sequential steps, something like a checklist. These steps have to be conducted in a specific order, following a pre-specified protocol. An example of an Administrative workflow might be the process of data-backup in a company. Details of the backup task could be that it has to take place at 1.00am every night, it should start with section A that has the most valuable information (e.g. payroll data), followed by section B that has the employees working data, move on with backing-up transactions of non-importance, etc. The main purpose of those workflows is to automate a well-defined cycle of operations that is quite simple and does require seldom modifications. Such workflows could be used for the automation, clearing and maintenance of indoor operations of a company since they do not deal with external stakeholders such as customers or other companies.

Ad hoc –weakly structured workflows

Finally Ad hoc workflows [Bates, 1993] present a different approach to the workflows seen so far. Ad hoc workflows reside at the area that the above categories do not. This category is related to repetitive processes that do not necessarily include a structured process and pre-defined actors. An example of the scope of Ad hoc workflows could be the compilation of a new document such as a sales report, a project proposal, a product evaluation report, etc. Such tasks could require actors across a wide range of departments and demand different skills. Their structure is usually vague and in some cases it is hard to be identified.

These workflows could be regarded wrongly as groupware, the predecessor of workflows, where many users were allowed to share information among them. However, groupware did not have any intelligent diversion of activities and it was not linked to a particular business process compared to the workflow.

As seen from the above workflows can have different types depending on their operational domain. Workflows can describe a process thoroughly as well as highlight the relevant operational aspects. Their advantages can be even more, offering leveraged perspectives to a business process, allow different diagrammatic operational views, etc. The following section refers to them in more detail.

2.3.2 Workflow Advantages

Workflows lead to automation of processes allowing their stakeholders to interact among them in a more effective, productive way. Workflows can define the necessary filter to discriminate among the productive and non-productive parts of a process. This can contribute significantly to areas such as the business process redesign and re-modelling depending on the current and/or any anticipated needs.

The workflow technology can actively promote this by automating the flow of information. More specifically workflows:

- *"They can automate the flow of informa*tion throughout the entire enterprise.
- They can integrate individuals, their roles and functions.
- They can easily be tailored to model the individual's work style and decision-making. Users only need to deal with the task at hand.
- They can formalise business procedures within your system. They do this by providing workflow tools that manage these procedures. This means you can control and audit your business *processes more effectively."* [Bates, 1993]

Workflows help in understanding how an enterprise works, identifying in parallel behavioural patterns and repetitive tasks. This results to better comparability and can represent efficiently the flow of actors in a process execution. This allows the detection of the non-productive activities that afterwards can be eradicated.

Since in modern businesses everything is about information, the workflow technology can provide the ability to bring the right information at the right time without spending valuable resources in harvesting it manually.

Since the executed processes lead to the generation of events it would not be wrong to say that workflows are event-driven. The tasks that take place within a certain system, triggered by the system users (actors), could be regarded as events (if they are simple) or a series of events based on their complexity.

While a system is operating, several tasks/events or series of events take place. If this is within the context of a workflow, the next step of the process could be envisaged and the task could be forwarded to the right person in order to continue the flow and lead to the completion of the task. Since the workflow is aware of the procedure that has to be followed there are fewer error margins. A reason for that is that the human factor is absent from the definition of the process and there is no interaction with physical means (objects that have to be transferred in order to move to the next task, letters that have to be sent, etc.)

2.3.3 Workflow Perspectives

Based on the workflow advantages, as seen in section 2.3.1, companies and organisations are using them to increase their productivity [Kueng, 2000], ensure their procedure efficiency and preserve the quality of their services. Procedural efficiency in a company can be ensured by the constant satisfaction of the business requirements. In order for this to be accomplished its business procedures have to change along with the given requirements and/or the anticipated problems.

Odgers et al. (1999) have identified three possible classes of problems that workflow systems may face. These are the Organisational Issues, the Retain of control whilst decentralising behaviour and the Load Balancing.

Organisational issues within a business can relate to the hierarchical structure that the business follows. This can be in terms of the production chain, communication with internal or external stakeholders and the allocation of responsibilities. Since these models differ from business to business there may be difficulties in exchanging product information, communicating with the organisation schemes of other companies (could be suppliers, customers or both), pinpointing potential problems or anomalies, etc. Due to the high range of differences in organisation schemes it is hard to establish a widely applicable workflow model, automate it and efficiently monitor it.

Workflows can run in several areas within one organisation taking into account its unique characteristics like: its context (e.g. operational domain), its procedures (e.g. quality procedures), its topology (e.g. different departments and their location), etc. Each workflow could have its own behaviour but the overall monitoring should be centralised. Monitoring can become problematic and insufficient due to the nature of the decentralisation (less control, loose project management, limited visibility of the overall picture). The need to overcome these problems seems essential.

Workflows can define the way processes, actors and resources should cooperate within an operational context. However, there is no indicator on how the workload is being distributed. This results to the so-called load balancing problem. Processes can get overloaded due to internal or external factors, having as a result the creation of prolonged task queues. If the queue exceeds a certain limit it may result to even unprocessed-task returns to their originators [Odgers et al., 1999].

The problem can be distinctive and hard to avoid in areas where a workflow is deployed within more than one organisation. An example could be the workflow for the manufacturing of an aircraft's engine where: The aircraft company may design the actual engine, send it for the components-assembly to a different one, outsource its testing to a second one, assign the payments to a third, etc. In such cases some parts of the workflow are regarded as external factors for the organisations, and therefore there can be limited control over them. This in terms of the load balance problem can have as a consequence the "*flow of the work not to be able to shut down*" [Odgers et al., 1999].

Having workflows to control the procedures denotes that they may need to be adapted or change in order to meet the on-going business requirements as well as the potential problems that may occur. Consequently a new need is being formulated, the need for intelligent workflow management. Via intelligent management the flexibility of a workflow system is increased, allowing fast, prompt responses to a constantly changing environment.

2.3.4 Workflow management with Petri nets

Towards the efficient definition, orchestration and control of processes, significant work has been conducted with frameworks based on Petri nets [Van der Aalst, 1998]. Petri nets, invented by Carl Adam Petri [Petri, 1962], have been used for the "the description and analysis of concurrent processes which arise in systems with many components" [Scholarpedia, 2008]. Petri nets are being composed by states and transitions that refer to substances and reactions accordingly. An example of a petri net used to represent a production net can be seen in figure 2-1.

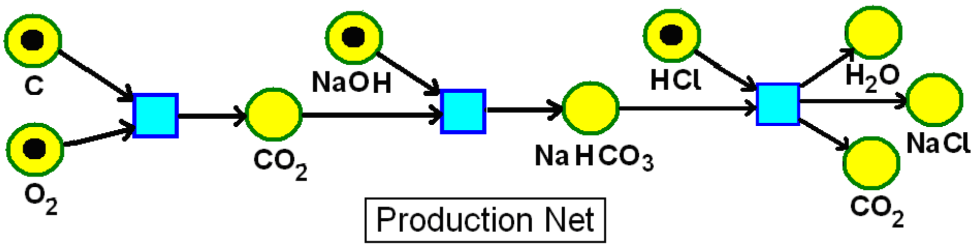

Figure 2-1: Example of a production net represented in terms of a petri net [Scholarpedia, 2008].

Another example of petri nets with relevance to the workflow complexity that they are capable to depict can be seen in figure 2-2.

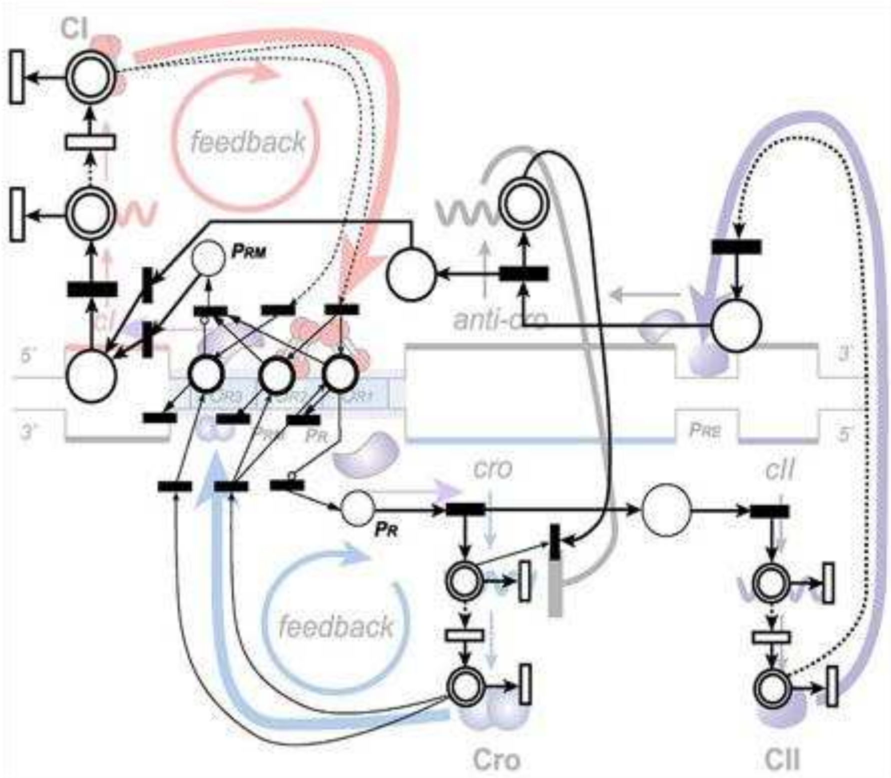

Figure 2-2: The central part of the switching mechanism of lambda phage and its hybrid functional Petri net (HFPN) description [CSML, 2010].

Petri nets have been used to model and analyse processes across a wide range of applications like hardware, embedded and distributed systems as well as the area of workflows. Van der Aalst & Hee (2004) indicate several advantages of Petri nets like their formal definition, graphical nature, expressiveness and subjectivity to a variety of analysis techniques. Their advantages, along with the fact that they are vendor independent, make them a powerful design language for the definition and representation of workflows.

Petri nets deal with workflow processes in a deterministic way [Baldan et al., 2005]. Workflow tasks, conditions and cases are being mapped as transitions, places and tokens respectively [Van der Aalst, 1998]. An example of how this is being conducted can be found within the context of a complaint workflow procedure mapping. The life cycle of a complaint could be the following: First the complaint is being registered and a questionnaire is being given to the person that made the complaint while the complaint is being evaluated. If the questionnaire is being returned within a pre-defined amount of time the complaint questionnaire is being taken into account otherwise its result is being discarded. Based on the outcome of the parallel evaluation the complaint is being forwarded in combination with the outcome of the questionnaire. The final outcome of the complaint is being checked and finally the complaint is being archived. Figure 2-3 shows the life cycle of the complaint depicted in terms of a Petri net.

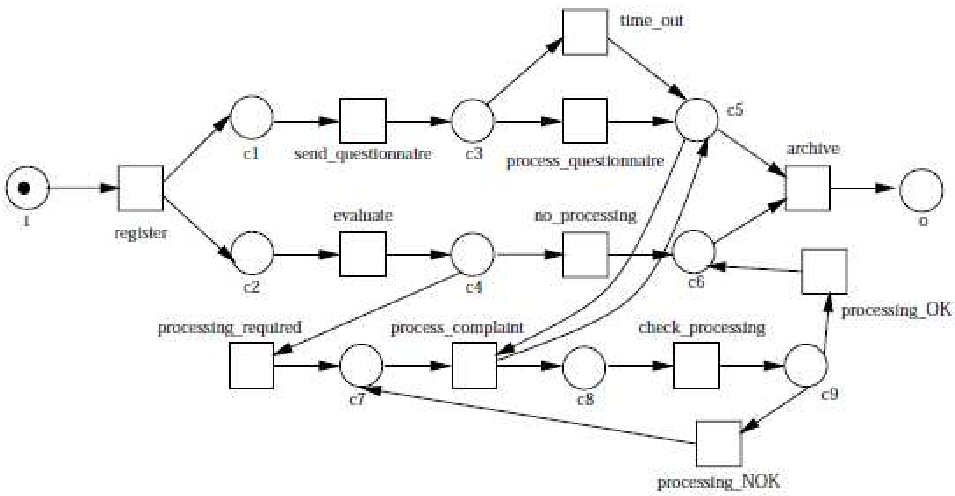

Figure 2-3: A Petri net for the processing of complaints [Van der Aalst, 1998].

Petri nets can represent single cases as shown from the figure above. However, they could even allow the representation of multiple cases within the same schema. This could lead to possible mixing of different cases, a potentially undesirable outcome since it can lead to information loss. In order to deal with such cases Petri nets offer the possibility to represent abstraction with the use of high level Petri nets. High level Petri nets contain nodes that contain extra information regarding the exact identity of them. Figure 2-4 shows the representation of a high level Petri net.

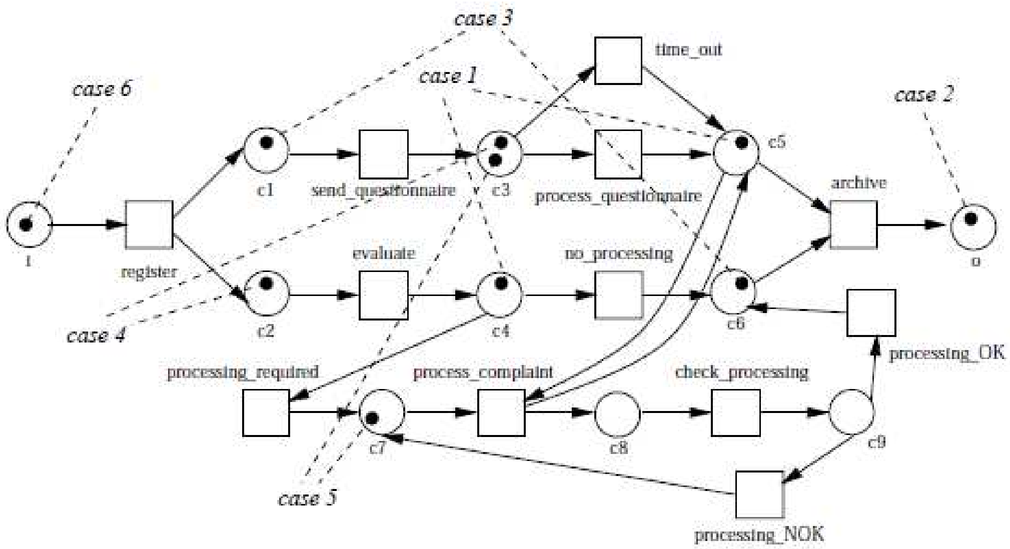

Figure 2-4: Tokens have a case identifier which allows for the separation of cases [Van der Aalst, 1998].

Petri nets used specifically for the modelling of isolated cases, or formally workflow process definitions, are referred to as Workflow Nets (WF-nets). A Petri net can be referred as a workflow net if it meets two requirements: “First of all, a WF-net has one input place (i) and one output place (o). A token in i corresponds to a case which needs to be handled, a token in o corresponds to a case which has been handled. Secondly, in a WF-net there are no dangling tasks and/or conditions. Every task (transition) and condition (place) should contribute to the processing of cases. Therefore, every transition t (place p) should be located on a path from place i to place o” [Van der Aalst, 1998].

Given characteristics like expressiveness and formalism can be used in workflow management systems in the form of high-level petri nets or colour petri nets. Several tools have used them towards this direction like Design/CPN [University of Aarhus, 2006] and ExSpect [EUT/D&T Bakkenist, 2000]. These tools, based on their foundation, can work along with the pre-specified algorithm that the petri net follows. However, several limitations exist where:

- patterns involve several instances,
- there are advanced synchronisation patterns
- there are cancelation patterns [Van der Aalst & Hoefstede, 2002].

Yet Another Workflow Language (YAWL) has been suggested as a more advanced high level petri net approach in order to overcome the previous limitations. YAWL supports a broader range of processes which traditional workflow management systems cover. However, still the foundation of this approach refers to an algorithm-structured foundation that a petri net provides. If this approach was to be applied to a monitoring system the system should be able to go beyond its substance algorithmic boundaries.

Although Petri nets seem a solid and promising approach towards the effective representation of workflows there is a mentionable drawback when used for the monitoring of business processes. More specifically if the undergoing process includes elements of uncertainty, unclear temporal relationships and vague knowledge regarding the process stages, the intelligent monitoring of a workflow seems beyond the context of Petri nets as stated above.

2.3.5 Intelligent Workflow Management

While the scale and complexity of business processes rises, the need for their effective management is being formulated. Workflows can be used to describe the different aspects of a process, automate it up to a certain extent and allow the involved stakeholders to interact in a more proficient way. However, due to the complexity of processes, workflow management is needed in order to deal with the existing uncertainty, the present risks in business operations and other unanticipated situations.

In order to have effective workflow management certain criteria should be fulfilled: the management should be resilient enough to changes, follow the dynamics of the operational environment and be effortlessly subject to active or passive monitoring. Traditional workflow monitoring faces limitations since it is restricted to sets of hard-coded rules. These rules are attached to certain event patterns that do not necessarily depict the on-going changes of a vigorous environment. An example could be the rectification of a process that operates in a humble way by reassigning roles, changing priorities and/or redistributing the workload [Stark & Lachal, 1995]. Another example can be seen within workflows with the present association of tasks and certain roles rather than tasks and individual participants [Stark & Lachal, 1995]. In this way the workflow decides who is responsible for a certain task based on his / her role, rather than taking into account any (possibly disruptive) present conditions of the participant (illness, provisional absence, etc.). As a result collisions and possible deadlocks may occur within the process.

Based on the current limitations posed by the traditional workflow monitoring systems, a more effective and intelligent way of monitoring workflows should be articulated. Monitoring should be flexible enough to follow the changes of the workflow environment, autonomous in terms of learning and collaborative with both the operational environment and its surrounding systems.

2.3.5.1 Architecture Technologies

Several technologies that pertain to the efficient management of workflows can be seen in the literature such as the object-oriented workflow management systems and the intelligent agents. This section will discuss on the above distinct technologies as well as several others which are up to a certain extent relevant to them, such as the OMG workflow facility, the CORBA and the WIBOs.

Object-oriented workflow management systems offer a customised way for users who come across them, since they model the workflow aspects in a granular (object) basis. Therefore users with no excessive knowledge can understand the concept behind a system in a natural way. Developers can also find familiar this approach since they are able to work and apply acquainted entities like those of polymorphism, inheritance and encapsulation, common in object-oriented languages. Development of an object-oriented workflow management system can be structured and subject to rapid changes without major alterations. Its object-oriented architecture significantly subsidises to that. The TriGSflow system [Kappel et al., 1995] is an example of object-oriented workflow management architecture. TriGSflow consists of an object-oriented database system that uses rules, objects and roles to facilitate a business process model. The system contains a rule-based model and supports object evolution in terms of roles. In that way the "flexible modelling and enactment of business processes is supported, allowing changes even during workflow execution" [Fakas & Karakostas, 1999].

Object-oriented approaches seem convenient in terms of implementation and usability. However, they do not present the necessary robustness when dealing with workflow execution problems.

A reason for that is that its major focus is on the implementation, deployment and structural maintenance. However, prerequisite for those is the fact that the operational domain will not change significantly. If a change or a sequence of changes occurs the system has to be re-engineered and revalidated.

Workflow Intelligent Business Objects (WIBOs) [Fakas & Karakostas, 1999] is another workflow management architecture which is based on the intelligence, autonomy and collaboration among workflows. WIBO architecture resides within the family of object-oriented workflow management. However, it has several extensions in order to overcome the main limitation of the object-oriented approach. Since they are objects they contain several generic characteristics which are afterwards specialised through restricting or enhancing their behaviour. The objects collaborate among them in order to enhance their efficiency and their mighty learning potential. Finally WIBOs incorporate a peer-to-peer approach in order to maximise their autonomous character by acting both as clients and servers while exchanging messages. Their hierarchy is shown in figure 2-5 below.

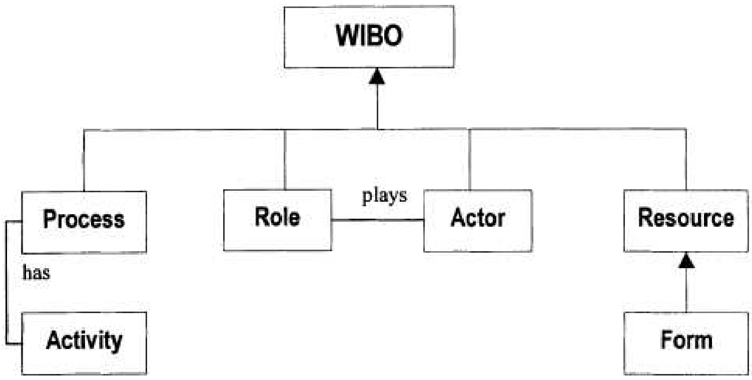

Figure 2-5: "The WIBO hierarchy" [Fakas & Karakostas, 1999].

OMG (Object Management Group) workflow facility contains a range of interfaces and appropriate semantics for the execution control, monitoring and interoperability of workflows. The interfaces can be defined separately for each workflow and “include their relationships and dependencies with requesters, assignments, and resources” [OMG, 2000].

A different architecture which is based on CORBA has also been proposed for workflow management systems. Its key aim was to allow management exchange of data between models (interworking) and the effective communication among Object Request Brokers (ORBs). The products of the above architecture have a main unit of an ORB “which enables objects to transparently make and receive requests and responses in a distributed environment” [The Open Group, 1999]. The products could also provide object services for their applications that could potentially share. These services could “support basic functions for using and implementing objects, as well as common facilities” [The Open Group, 1999]. The range of the CORBA architecture is quite wide from full centralisation to full distribution. CORBA although powerful in terms of the provided functionality has stated several problems in terms of the persistence of objects [Silva Filho et al., 1999]. As a result the standard CORBA references could take a very long time in terms of the development and release phases.

Intelligent agents on the other hand offer a more revolutionary approach to the problem. Agents are software-based computer systems which as Wang et al. (2005) denote they have the following characteristics:

- Autonomy (they can operate without human coordination or intervention)
- Social Ability (they can communicate with other systems, agents)
- Reactivity (agents can respond to changes that occur within their operational context)
- Pro-activity (they could take initiative within their environment and behave according to their pre-specified behaviour) [Wooldridge, 2009]

Agents are instructed to perform certain tasks within a specific environment for which they have some sort of knowledge. They have specific capabilities and a pre-given attitude (behaviour) towards their goals. In order to achieve their goals they have to use their knowledge, generate a solution plan and then execute the plan [Wang et al., 2005]. Agents can collaborate with other agents within the context of a multi-agent environment in order to overcome the limitations posed as single units and achieve higher goals.

Referring to the workflow limitations, as stated in section 2.3.5, the usage of intelligent agents could be of benefit [Jennings et al., 2000; O'Brien & Wiegand, 1998]. Agents could be used towards the decentralisation of the workflow control, reactivity, efficient resource management, easy interaction and firm decision making. The literature can show several examples where the agent technology has been integrated with workflows in order to assist the efficient management of business processes. The Agent-based Process Management System (APMS) architecture [O'Brien & Wiegand, 1998] is one example of applied agent technology which has been used to extend the intelligent management of workflows. APMS has been used in the projects of ADEPT and BeaT [O'Brien & Wiegand, 1998] using multiple autonomous agents in order to achieve the automation of the process resources, the decentralisation of organisational structures and the adaptation to environment changes. The adoption of APMS from organisations, although promising, was accompanied by a significant cost. Agent Enhanced Workflow (AEW) proposed by Judge et al. (1998) combined a layer of agents "given responsibility for the provisioning phase of business process management, whilst the underlying workflow system handles process enactment" [Odgers et al., 1999]. Mainly AEW worked as an interface between workflow systems and other software tools for the needs of their monitoring and control. Another example of using multi-agent system within the context of E-Commerce was presented by Chen et al. (2000). The unique characteristic of that system was that its multiple agents were able to change their behaviour dynamically based on the operational domain.

Intelligent agents can be used in a very diverse range of cases. An important advantage they offer is their possible combination in order to be able to solve different problems. Such combination can result to complicated working units as well as architectures. Figures 2-6 and 2-7 below show two examples of an intelligent agent and architecture of multi-agents respectively.

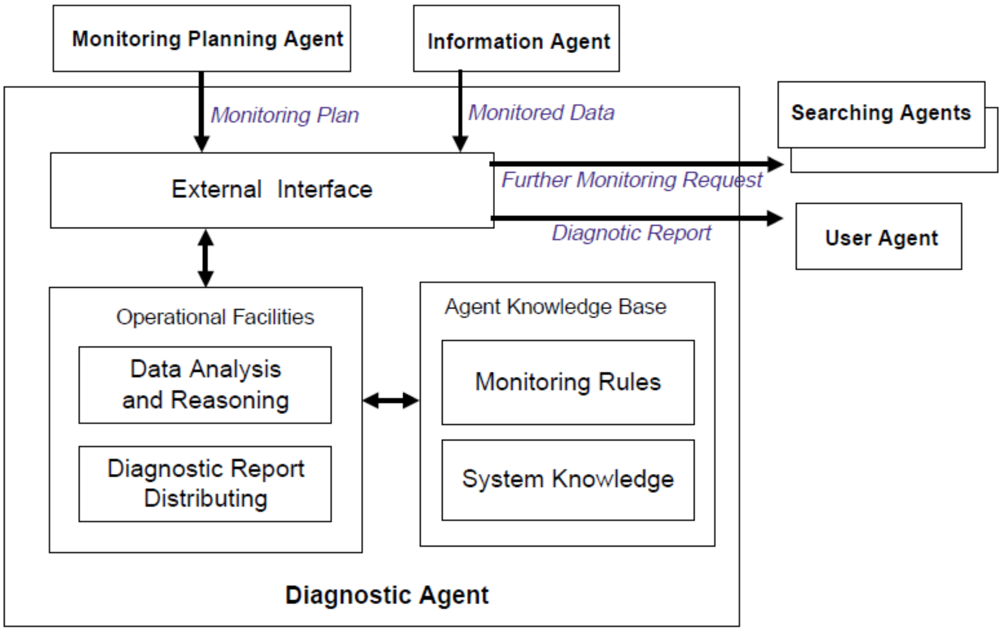

Figure 2-6: Structure of an intelligent agent [Wang et al., 2005].

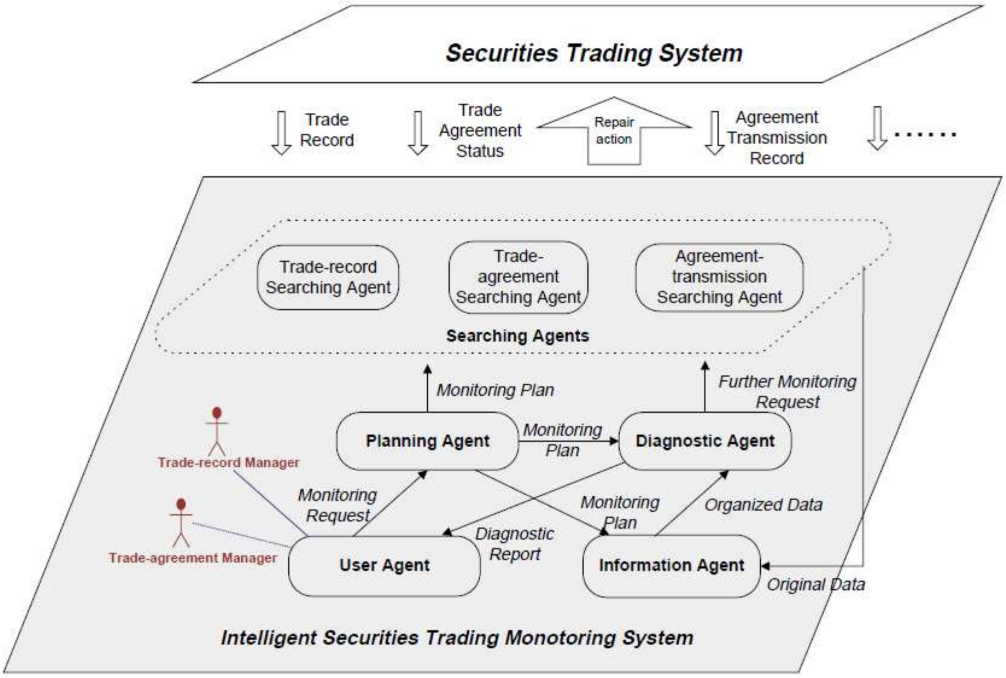

Figure 2-7: Architecture of an intelligent multi-agent system [Wang et al., 2005].

Intelligent agents can be flexible over a wide range of applications and be a solution to different problems. However, this approach is not followed in this thesis since they pose disadvantages regarding the security of an investigated system, the lack of standards in terms of their communication and the extensive work that is required for their training and subsequent improvement of their “intelligence”.

2.3.5.2 Challenges

The challenges in intelligent workflow management as stated by Odgers et al. (1999) could be summarised as the dynamic process creation, user centred workflow and the knowledge management tools for process management. Starting from the dynamic process creation, the complexity of the modern customer demands dictate for dynamic business process. These processes should be able to adapt and be personalised successfully to the needs of the current market.

Workflows should be user-centred and disassociated from their current role which is usually restricted to workload allocation and performance measurement. The workflow actors of each organisation should be taken into account while imposing a workflow management in order for it to be acceptable, usable and sustainable.

Finally each organisation should be able to learn and improve from its own procedures and processes. The meaningful interpretation of the captured data as well as the information distillation (in order to be able to extract valuable knowledge) constitute an on-going need while managing the workflows of an organisation. Proactive knowledge management tools are required in order to achieve an effective process management.

2.4 Temporal Logic

For the effective monitoring of workflows, their execution traces or the active execution instances need to be examined. Usually workflow operations leave traces in the form of a timed event sequence. These sequences are often being stored in an accessible region and they could be analysed afterwards for security purposes, forensic investigations, auditing surveys, etc. After a certain elapsed time span these event sequences are getting archived.

Traces from workflow executions could be used either at live mode, which indicates a real-time usage of the generated data or at post inspected mode. The latter indicates that all the operations have taken place; data were produced and processed afterwards. These traces seem a valuable source for the reconstruction of the workflow execution path within the scope of a business process investigation. As a result an appropriate time theory is being required that can represent events with the necessary precision. Meanwhile the representation of events along with their related temporal relationships is an essential requirement. Having the data along with their temporal information formulates a new requirement: the need for pattern discovery identification within their temporal repositories. Pattern discovery is of high importance since it can indicate the patterns that can present either a problematic or a healthy state of a workflow execution respectively.

The analysis of the temporal information can be of significant importance in domains where the diagnosis is prominent. This could be in areas that deal with behaviour analysis, web navigation, network monitoring, financial transaction monitoring, fraud prevention and diagnosis, etc. Generally the temporal analysis is imperative towards the understanding of cognitive processes that include human interaction. One approach that seems similar to the knowledge discovery from workflow traces is the knowledge mining from interaction traces, since it aims to extract patterns from a temporal repository. Knowledge mining from interaction traces is discussed further in the following section.

2.4.1 Knowledge Mining from Interaction Traces

The knowledge extraction from a temporal repository can be a highly important task in pursuit of the accurate information retention. The processes that incorporate human resources are heavily reliant on such information since at certain cases it is the only way to track, inspect and improve a variety of processes. As a result the efficient processing of any available temporal information is a concept that attracts significant attention.

The analytical approach of the temporal information in a particular domain follows a series of steps. While conducting analysis the focus is on mining interesting patterns from the available behavioural knowledge. As a first step towards the mining of patterns, the available information should be gathered from the inspected domain [Fisher & Sanderson, 1996]. In areas where the interaction among users and their application area is being recorded, the constitution of the desired temporal repository can be a plausibly simple task. However, this is not the case in areas that actions may occur outside the context of the system.

Existing literature shows several pieces of work on the knowledge discovery from interaction traces [Fisher & Sanderson, 1996; Hilbert & Redmiles 2000]. Work on interaction traces has been conducted with the focus on extracting knowledge from human activities, a process that has similarities with the knowledge extraction from workflow traces. However, there is a difference among them in terms of the quality of the traces as well as the operational domain of interaction traces. An indicative example could be the interaction among car drivers, something that was actually used from the Fisher & Sanderson (1996) for the knowledge mining.

The conducted analysis was focused on the extraction of behavioural interaction patterns by using SPARQL [Prud'hommeaux & Seaborne, 2008] request language. SPARQL can be used to query data represented in RDF format. Results extracted from SPARQL queries were able to be compacted enough in order to formulate a new piece of a rather abstract but formal description. The acquisition of a new formal description establishes a new piece of knowledge that could be added to the knowledge repository. Towards that direction ABSTRACT [Georgeon, Mille & Bellet, 2006] has been developed, a knowledge discovery tool that can calculate the existence of interaction instances. However, these calculations do not indicate a clear pattern discovery.

In the area of knowledge discovery, work has been conducted by introducing trace mining techniques towards the enhancement of tools like ABSTRACT. These mining techniques include algorithms that can iteratively be used to assist the requests of analysts. Mining techniques used on traces have been described thoroughly by Cram et al. (2008). Cram et al. have conducted a significant amount of work on knowledge discovery at real time, based on interaction traces [Cram, Mathern & Mille, 2011]. The above mentioned research work has been focused in the constitution of a framework that attempts to discover patterns among chronicles [Dousson, Gaborit & Ghallab, 1993] (The term “chronicles” can refer to any time series of ordered events).

2.4.2 General Time Theory

While business processes are being executed several events take place in the form of a sequential series. The time among them and their sequential order is of high importance. More specifically it is considerably significant:

- whether an event A had happened before, after or at the same time as an event B
- whether there was any interference among them or with other events
- whether their duration is known or not.

An example that indicates the significance of the sequence of events can be borrowed from legal reasoning. Usually in legal case proceedings several witnesses are summoned to testify in court. The sequence of events as described from each witness, their relationship with other events and their actual duration of the intervals between events, can be of critical importance to prove or disprove some facts in the legal case.

In order to be able to represent the time in cases like the above, along with other natural phenomena and activities several theories have been developed. The literature can show three distinct choices [McDermott, 1982; van Benthem, 1983; Allen & Hayes, 1989, Berndt & Clifford, 1996; Ma, 2007] which they differentiate in terms of their selected primitive time unit. Their inspected time elements, which essentially define the whole theory, can be distinguished into points that represent a certain snapshot in time, intervals that represent specific time duration and a combination of both the above.

A brief example of a time representation with points could be the spaceship launch which is supposed to take place at 15.36.30 EST. An example with intervals could be the announcement some time afterwards that the spaceship has successfully reached the upper levels of the atmosphere, 1.5 hours after its successful launch.

2.4.2.1 Point-based Systems

The point-based systems, as their name indicates, are being represented as an ordered set of points: (P, ≤). P represents a set of points and ≤ indicates a relation that orders the set partially or totally. In systems where the time representation is via a point-based system, intervals are being represented either with sets of points [McDermott, 1982; van Benthem, 1983] or with pairs of points [Bruce, 1972; Shoham, 1987; Halpern & Shoham, 1991; Ladkin, 1992] that follow a specific order.

A problem that derives when defining intervals as objects, as several researchers have indicated is the so–called **Dividing Instant Problem** [van Benthem, 1983; Allen & Hayes, 1989; Vila, 1994; Ma & Knight, 2003]. The Dividing Instant Problem derives when trying to represent a boundary point that is in between of two successive intervals. An example given by van Benthem (1983) describes the problem in an illustrative way:

A fire that had been burning was later burnt out.

Upon the above statement two states can be defined:

The fire was burning and The fire was not burning which respectively hold true throughout two successive point intervals. The intervals can be defined based on the definition of point-based systems as $< p_1, p >$ and $< p, p_2 >$. Based on the previous interval representation the system is called to answer to the following question: Was the fire burning or not burning at point p? (**Question 1**)

This, if referred to the open/closed nature of the point-based intervals, it can be reformulated as to which set between $< p_1, p >$ and $< p, p_2 >$ is open/closed at the investigated point p. The possible cases in this situation are the following:

“(a) The fire was burning rather than not burning at p;

(b) The fire was not burning rather than burning at p;

(c) The fire was both burning and not burning at p;

(d) The fire was neither burning nor was it not burning at p.” [Ma, 2007]

From the above assumptions, options **c** and **d** violate the Laws of Contradiction and Excluded Third mentioned by van Benthem (1983), therefore they are ignored. However, choices **a, b** remain without providing enough reasoning as to which is true as an answer to the **Question 1** above. As a result since there is a margin for case misclassification and possible confusion, the approach of point-based systems has been characterised as indefensible and unsatisfactory [van Benthem, 1983; Vila, 1994; Ma, 2007].

Strictly speaking point-based cannot depict time elements in a changing environment that deals with uncertainty since a lot of chronological information remains unknown. These can be systems in real life where the start/end of a time event is roughly known or completely undefined. An example could be something like John had a coffee break with Stef before Gabriela coming in or During the morning there was a power outage that lasted for two hours, etc.

2.4.2.2 Interval-based Systems

Time intervals were suggested as an alternative approach to the point-based systems. An important detail, present in both point-based and interval-based systems, is that points and intervals constitute the most granular elements of a system respectively. Time intervals seem a more descriptive way in representing time scenes from the real life like examples of John was playing his guitar the whole afternoon.

If intervals are being taken as primitive temporal units, points in time can be represented as “maximal nest of intervals that share a common intersection” [Ma, 2007]. Alternatively they can be represented as meeting places of intervals [Allen & Hayes, 1989, Berndt & Clifford, 1996].

Allen has proposed his temporal theory [Allen, 1984; Allen & Hayes, 1989] based on the interval-based approach. Since intervals are the primitive entities Allen used several temporal relations to define the relationships among them. These relations were thirteen in number and were the *‘After’, ‘Before’, ‘Contains’, ‘During’, ‘Equal’, ‘Finished-by’ , ‘Finishes’, ‘Meets’, ‘Met-by’, ‘Overlapped-by’, ‘Overlaps’, ‘Starts’* and *‘Starts-by’*.

By introducing the above approach Allen managed to overcome the difficulties stated by the Dividing Instant Problem. The reason for that is that in the Interval-based approach there is no need to classify points in terms of whether they belong or not to a certain interval. According to Allen points in time are fading, since there are not entities where things happen or are true. Therefore they were excluded from the temporal ontology overall.

Such an approach although having advantages, it seems inappropriate to systems that face continuous change [Galton, 1990]. On top of that there is no negation that certain events take place unexpectedly, literally at no time. Therefore it necessary to have time points which should be used for temporal reference.

Ma & Knight (2003) give an illustrative example of the interval-based theory limitations:

A ball was thrown into the air from the east to the west.

Following the route of the ball there are certain moments where it is hard to define its position in terms of an interval. The moment that the ball was at the east and just below its apex is one of them, which was immediately followed by the state that the ball was at its apex. That particular moment was immediately followed by the state that the ball was at the west side and just below its apex.

The point which the ball was at its apex, although trivial in terms of time duration, is an example that should be represented as a point of zero duration, rather than an interval or moment [Allen & Hayes, 1989]. That specific moment was actually the only one during the ball's orbit that its speed became zero.

Interval-based time theory seems sufficient for a number of temporal representations but as Galton (1990) indicated it faces some weaknesses in terms of clarity in semantics and completeness.

2.4.2.3 Point & Interval based systems

Points-based time theory and interval-based time theory deal successfully with some aspects of the temporal representation but they both fail to represent successfully the whole range of temporal aspects. Ma & Knight (1994) proposed a general time theory where both points and intervals are being considered as temporal primitives. General time theory regards interval-based and point-based time primitives as equal. As a result points are not defined as limits of intervals and equivalently intervals do not consist of points.

Another characteristic of the general time theory is that it reduces the number of the 13 time relations defined by Allen & Hayes (1989) to just one: the Meets relationship. The general time theory takes both points and intervals as primitive and consists of sequences of triads. Any triad contains:

- ❖ a time element (**T**) which is a non-empty set of time elements;
- ❖ a **Meets** relationship among T and other elements which is a binary order relation over T;
- ❖ a duration (**Dur**) which is represented as a non-negative number and is a function from T to R_0^+, the set of non-negative real numbers.

A time element t is called an interval if Dur (**t**) > 0; otherwise **t** is called a point.

The basic axioms concerning any triad (T, Meets, Dur) are stated below:

A1. $\forall t_1, t_2, t_3, t_4$ (Meets $(t_1, t_2) \wedge$ Meets $(t_1, t_3) \wedge$ Meets $(t_4, t_2) \Rightarrow$ Meets (t_4, t_3))

Axiom **A1** indicates that if "a time element meets two other time elements, then any time element that meets one of these two must also meet the other. This axiom is actually based on the intuition that the *"place" where two time elements meet is unique and closely associated* with the time elements" [Ma, 2007].

A2. $\forall t \exists t_1, t_2$ (Meets$(t_1, t) \wedge$ Meets(t, t_2))

"That is, each time element has at least one immediate predecessor, as well as at least one immediate successor" [Ma, 2007].

A3.$\forall t_1, t_2, t_3, t_4$ (Meets $(t_1, t_2) \wedge$ Meets $(t_3, t_4) \Rightarrow$

Meets $(t_1, t_4) \nabla \exists t'$(Meets $(t_1, t') \wedge$ Meets (t', t_4)) $\nabla \exists t''$ (Meets $(t_3, t'') \wedge$ Meets (t'', t_2)))

where ∇ stands for "exclusive OR".

"That is, any two meeting places are either identical or there is at least a time element standing between the two meeting places if they are not ident*ical"* [Ma, 2007].

A4.$\forall$ t_1, t_2, t_3, t_4 (Meets (t_3, t_1) $\wedge$ Meets (t_1, t_4) $\wedge$ Meets (t_3, t_2) $\wedge$ Meets (t_2, t_4)) $\Rightarrow$ $t_1 = t_2$)

Axiom 4 indicates that the time element between any two meeting places is unique.

A5.$\forall$ t_1, t_2 (Meets (t_1, t_2) $\Rightarrow$ Dur (t_1) > 0 $\vee$ Dur (t_2) > 0)

Axiom 5 indicates that time elements with zero duration cannot meet each other.

A6.$\forall$ t_1, t_2 (Meets (t_1, t_2) $\Rightarrow$ Dur ($t_1 \oplus t_2$) = Dur (t_1) + Dur (t_2))

"*That is, the "ordered union" operation over time elements is* consistent wi*th the conventional "addition" operation over the duration assignment function, i.e., "Dur"*" [Ma, 2007].

For the general time theory T introduced above, the following two assumptions are made from real number theory:

Assumption 1. "The set of real numbers is totally ordered by the less-than-or-equal-*to relation '≤', where '>' is the "bigger than" relation, that is, not '≤'* " [Ma, 2007].

Assumption 2. " *'+' is the conventional addition operator over (non-*negative) real numbers" [Ma, 2007].

Regarding time relations, all Allen's temporal relations can be reduced to the 'Meets' one. This reduction is being shown in the following statements:

1. Equal $(t_1, t_2) \Leftrightarrow \exists t', t'' (\text{Meets}(t', t_1) \wedge \text{Meets}(t', t_2) \wedge \text{Meets}(t_1, t'') \wedge \text{Meets}(t_2, t''))$

2. Before $(t_1, t_2) \Leftrightarrow \exists t (\text{Meets}(t_1, t) \wedge \text{Meets}(t, t_2))$

3. Overlaps $(t_1, t_2) \Leftrightarrow \exists t, t_3, t_4 (t1 = t_3 \oplus t \wedge t_2 = t \oplus t_4)$

4. Starts $(t_1, t_2) \Leftrightarrow \exists t (t_2 = t_1 \oplus t)$

5. During $(t_1, t_2) \Leftrightarrow \exists t_3, t4 (t_2 = t_3 \oplus t_1 \oplus t_4)$

6. Finishes $(t_1, t_2) \Leftrightarrow \exists t (t_2 = t \oplus t_1)$

7. After $(t_1, t_2) \Leftrightarrow$ **Before** (t_2, t_1)

8. Overlapped-by (t1, t2) ⇔ **Overlaps** (t2, t1)

9. Started-by (t1, t2) ⇔ **Starts** (t2, t1)

10. Contains (t1, t2) ⇔ **During** (t2, t1)

11. Finished-by (t1, t2) ⇔ **Finishes** (t2, t1)

12. Met-by (t1, t2) ⇔ **Meets** (t2, t1)

The completeness of the 13 possible exclusive order relations (the 12 stated above plus the 'Meets' relation) between any two time elements can be simply characterised by a single axiom as below:

$\forall t_1, t_2$ (Equal$(t_1, t_2) \vee$ Before$(t_1, t_2) \vee$ After$(t_1, t_2) \vee$ Meets$(t_1, t_2) \vee$ Met-by$(t_1, t_2) \vee$ Overlaps $(t_1, t_2) \vee$ Overlapped-by $(t_1, t_2) \vee$ Starts $(t_1, t_2) \vee$ Started-by $(t_1, t_2) \vee$ During $(t_1, t_2) \vee$ Contains $(t_1, t_2) \vee$ Finishes $(t_1, t_2) \vee$ Finished-by (t_1, t_2))

Also the exclusiveness of these 13 order relations needs to be characterised by 78 axioms of the following form:

$\forall$ t1, t2 (!Relation1 (t1, t2) $\vee$!Relation2(t1, t2))

where Relation1 and Relation2 are two distinct relations from the above 13 relations.

2.4.3 Temporal Logic on Workflows

When business processes are being executed, several events take place which can be represented in terms of a workflow. Recent work in the area has provided a logical foundation to define the events deriving within the execution of a workflow in terms of their temporal correlation.

Petridis, Ma & Knight (2011) have provided a firm foundation for business process modelling based on the points & intervals general time theory [Ma & Knight, 1994]. Within this representation approach the basic workflow entities: actions, events and processes are reduced to actions and events. The basic aspects of this representation are stated below.

An action has two folds: “1) an action name that identifies a certain type of action; and 2) an action instance that refers to an actual distinct performance of the action at a particular time” [Petridis, Ma & Knight, 2011]. Action names describe a certain type of non-instantaneous activity e.g. play basketball, drive a car, etc. A provided type of action can take place once, more than once over time, or may not take place at all.

Action instances are pairs consisting of an action (a) and a time moment (t). Action instances could be represented in short form as ai_s. For their formalisation it can be written that ai = (Name (ai), Time (ai)) where Name is a function from a set of action instances (AI) to a set of actions names A. Time is a function from the set of AI to the set of time moments M. Action instances are distinct and in order to denote that an action instance takes place over a time moment t the temporal proposition Performs can be used from reified temporal logic [Allen, 1984; Ma & Knight, 2001] e.g. Performs (ai, t).

Following the terminology of action and action-instance above the definitions of event and event-instance can be introduced. An event name is an identifier that can refer to a certain type of instantaneous activity, e.g. start a game, switch the engine on, etc. The set of events can be referred to as E and the events can be denoted as e_1, e_2, e_3... etc. Events like actions can take place once, more than once, or not at all.

Event instances as action instances can be represented as a pair of an event name and a time point e.g. (e, p) where $e \in E$ and $p \in P$. Event instances can be represented as ei_1, ei_2... etc. The set of event instances can be represented as EI. Each event instance can be written as ei = (Name (ei), Time (ei)), where Name is a function from the set of event instances EI to the set of event names E. Time is a function from the set of event instances EI to the set of time points P. Temporal proposition Occurs, e.g. Occurs (ei, p) can be used from reified temporal logic [Allen, 1984; Ma & Knight, 2001] to represent that an event instance ei occurs at a time point p.

Event instances are distinct, thus for any event instance ei can be imposed that "there are two special events associated with any action, through the corresponding time moment, i.e. the instigation event and *the termination event."* [Petridis, Ma & Knight, 2011] For any provided action a, the above statement applies indicating these two events as: instigate-event (a) and termination-event (a).

Equivalently to the definition of events, actions and their relevance to instances, the definition of a business process is given. A business process name is a set of a certain type of the business process, e.g. the general university enquiries process. A business instance is a set of action instances and event instances. For example the university's enquiries process taking place for the academic year 2012 – 2013, consisting of 6 distinct stages and described in its auditing handbook, is a business process instance.

Within the approach of Petridis et al. (2011) it is possible to define a temporal model TM for a business process (Pro). The process can be defined as the minimal set of temporal facts about action times, i.e. facts which remain true in each business instance of the business process.

Common temporal facts do not usually contain exact times since those cannot be usually common among a certain range of examples. Usually the set of temporal facts are usually given in terms of incomplete temporal knowledge. The set can contain duration knowledge but seldom could specify exact actual durations. An example of a temporal model could be the famous pin factory described by Adam [Smith, 1977]. Figure 2-8 shows the temporal model of the pin factory.

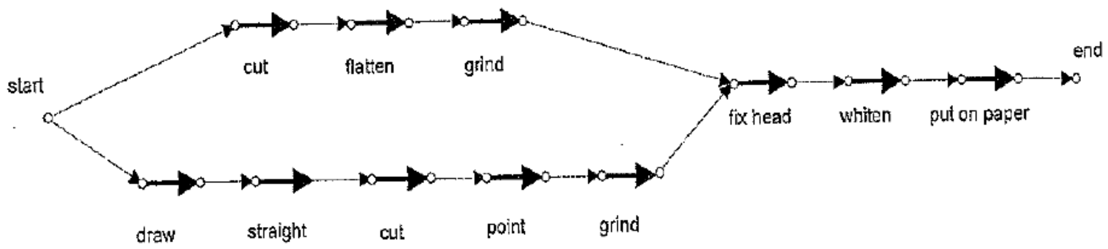

Figure 2-8: "The temporal model of the pin factory" [Petridis, Ma & Knight, 2011].

As it can be seen from the figure above there are no absolute times in the model. There is only presented the temporal order of the actions. This is not necessarily the general case as it can be seen from the soft-boiled egg process at figure 2-9. The depicted model indicates that the take out the egg action is between 3 minutes and 3minutes, 15 seconds after the egg is placed into the water. This duration applies to all instances of the process.

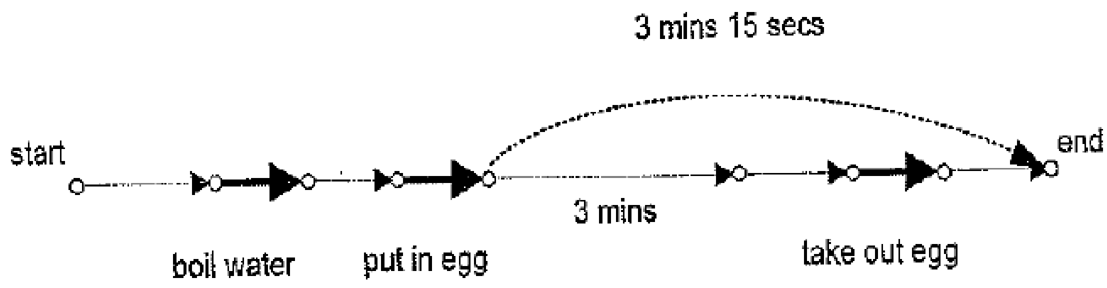

Figure 2-9: "The temporal model for soft-boiling an egg" [Petridis, Ma & Knight, 2011].

Based on the temporal model presented above a sub-process of the actual business process was defined. Actions and sub-processes share many characteristics. For example an action is an indivisible granular activity which can be assigned to a separate actor responsible for its completion. Once it is started, its finishing point is expected with no other interferences of other actions waiting for completion or commencement. From the perspective of the temporal model an action is only connected to the model via its start and end vertices.

A sub-process can be defined as above although a slight generalisation should be made: a sub-process should be any sub-graph of the temporal model which is only connected to the model via a start and end vertex. The definition of a sub-process of a temporal model TM can be as "any (non-trivial) sub-graph TM_1 of TM which can be disconnected from T by removing all in-arcs to a start vertex and all out-arcs from an end vertex" [Petridis, Ma & Knight, 2011]. Figure 2-10 shows a sub-process Pro_1 which is disconnected by removing in-arcs to vertex v_1 and out-arcs from vertex v_2.

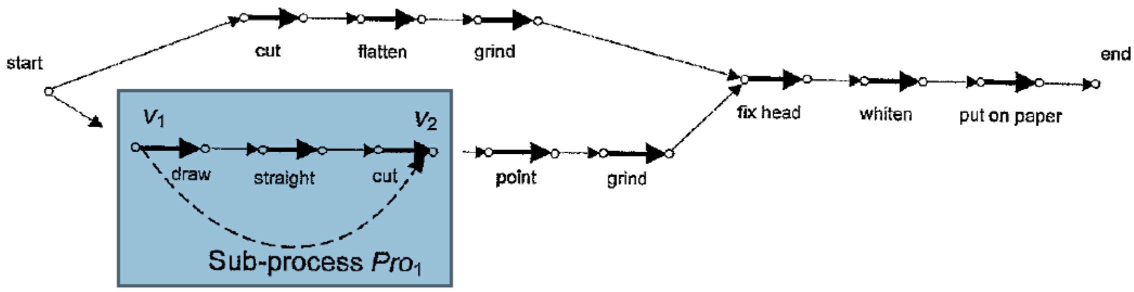

Figure 2-10: "Disconnected sub-graph defined by v1 and v2" [Petridis, Ma & Knight, 2011].

Actions in a temporal model TM can automatically be sub-processes to TM, so actions are also sub-processes. However, a difference between them is that sub-processes may be assigned to many actors towards their completion, whereas actions must be assigned to a single actor.

By using sub-processes in a temporal model the overall model can be depicted as a collection of the major sub-processes and their related intervals. The details within the sub-processes can be shown as separate diagrams afterwards.

Workflows can be the connection link among the definition of business processes and their actual execution. However, this is not enough in the case of the intelligent monitoring. The possibility this to be conducted via artificial intelligence and thus Case-based reasoning is been investigated. CBR provides a natural approach towards the efficient reuse of past available experience. The next section will provide a critique on CBR in comparison with other technologies as well as show the existing work in the area.

2.5 Case-Based Reasoning

When systems become larger in size and more complex in structure, Artificial Intelligence (AI) is being considered as a means towards their automation, effective management and coordination. AI unifies the underlying infrastructure, the designed models and the necessary machine intelligence in order to manage resources and workload efficiently. AI techniques are being used extensively in the areas of finance, engineering, medical field, military and education towards the solution of definite problems.

Among the AI techniques the Rule-Based Systems (RBSs), can also be referred to as Rule-Based Expert Systems (RBESs), and the Artificial Neural Networks (ANNs) are the most prominent towards the development of applications. These techniques, although common and quite widely familiarised, do present some hard to avoid downsides.

Starting from the RBESs, and briefly investigating their application model, it is observed that in order for a RBES to work efficiently a set of rules has to be defined that depicts the overall domain model. As a result a RBES system in order to be able to work, the whole domain knowledge has to be gathered in advance. This can be a difficult task for systems that are in operation for a number of years. Existing experience should be transferred into the RBES via rather simplistic rules, something that cannot be easily done if the system covers a wide operational spectrum. The number of rules in such cases would be massive. The maintenance of such a system could be proven a stiff task too, if the rules constantly change or new rules emerge, not to mention rules that contain overlaps among them. Above all if there is no defined rule for a given problem there is no possibility to find a solution for this particular problem [Kersten & Meister, 1996].

ANNs do not need the overall knowledge acquisition from the domain model in order to propose a solution to an addressed problem, an advantage compared with the RBSs. However, they do face remarkable limitations since they are numeric restricted [Arditi & Tokdemir, 1999] and their way of operating remains hidden while producing the solution. As a result a provided solution cannot be easily validated since critical information is missing like: the technical information, the parameter weights, the learning rules and the internal functionality of the ANN. Therefore explanation cannot easily be provided, making difficult their applicability overall.

Case-based Reasoning (CBR), a modern computational model, could be proposed towards the eradication of the above mentioned drawbacks. CBR is fast in construction compared to RBESs, easier to maintain and can cope with complex structures. This is an advantage in comparison with ANNs that use numeric input or symbolic patterns to deal with the complexity of structures. CBR can also work with a small number of cases, an advantage compared to the domain-knowledge acquisition requirements of the RBESs [Prentzas & Hatzilygeroudis, 2007] and the exhaustive domain requirements of the ANN learning phase [Jha, 2007].

Business process analysis is a complex domain with little explicit knowledge and a few reliable models. CBR can provide a way to reason with complex temporal events and episodic information coming from the automated data monitoring of workflow execution. Preliminary discussion with domain experts has shown that practitioners reason about workflow problems by recalling past occurrences and experience. This by default is the operational approach of a CBR system.

Throughout the following sections an introduction to the CBR methodology will be presented, its application on industrial systems will be shown as well as its connection with the workflows. Finally the entities of uncertainty in CBR, the adaptation and the temporal concept will follow along with the final conclusions of the chapter.

2.5.1 CBR Methodology – Architecture

AI systems seem an effective approach towards the management, analysis and support of large scale enterprise systems. Case-based Reasoning is an AI series of techniques that work towards the solutions of problems in a similar way to the human brain, based on experience from the past. Ian Watson (1997) in his book Applying Case-Based Reasoning: Techniques for Enterprise Systems states: “Case-based reasoning (CBR) is an intelligent-systems method that enables information managers to increase efficiency and reduce cost by substantially automating processes such as diagnosis, scheduling and design”. As Watson indicates CBR can be regarded as a computational methodology rather than a technique. CBR can use a variety of AI techniques towards the solution attainment during its Retrieval and Adaptation phases.

There are innumerable examples in real life that show how experts in technology fields work: familiarise themselves with the domain, collect specimens, define a methodology, etc. These techniques could be generalised. A professional in the medical field collects the symptoms of an illness and based on the knowledge gathered from the past can presumably identify the possible cause of them. Based on that past knowledge a medication can be formed and if the illness seems to be confronted, assurance comes and the rest of medication is being formulated with certainty. In the case of a not successful treatment the medication changes towards the next possible illness that the experience of the professional indicates, based always on the symptoms observed.

Professionals in the fields of law, mechanical engineering, industrial environments and even ordinary people in everyday life find solutions to problems based on analogy making [Hofstadter, 1985; Mitchell, 1993]. CBR is based on the experience acquired throughout the years rather than the overall knowledge available in the investigated domain. As a result its cases contain either problems with known solutions or problems with failed attempts to solve them. In both cases the user of a CBR system is of benefit since explanation is provided. As a result s/he knows what to do if the given solution was successful or what to avoid in any other case.

2.5.2 Case-Based Reasoning Cycle

CBR is a series of artificial intelligence techniques that uses existing (past) knowledge in order to find a possible solution to a given problem. This knowledge can be stored in repositories and formed in distinct cases. In CBR, problems are characterised from regularity and recurrence [Aamodt& Plaza, 1994]. Regularity indicates that the experience gained from one problem can probably be used in a similar problem, if this problem reappears. Reoccurrence refers to the fact that problems seem to reappear over time either regularly or periodically. Such problems can be categorised based on their type. The solution that works for one of them can possibly work, after applying the necessary changes, to another problem of the same type.

CBR techniques can be generalised in a formal way into four-step processes [Aamodt& Plaza, 1994]. This is usually referred to as the CBR cycle or *the four R's* processes where R's stand for Retrieve, Reuse, Revise and Retain. The operation conducted by each step deals with knowledge memory represented in the form of cases. A case in CBR can usually contain a problem and a known solution for that, as well as the relevant information that led to the addressed solution.

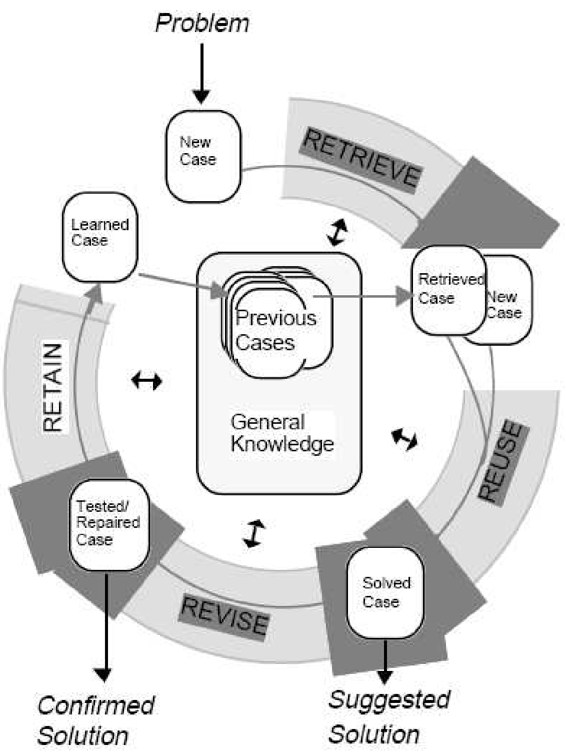

Figure 2-11: The CBR Cycle [Aamodt & Plaza, 1994].

The steps of a CBR process are explained briefly below.

1. **Retrieve**: When a problem is forwarded to a CBR system, the system regards it as a new case - problem with an unknown solution. The task of the CBR System, while examining a new case, is to search the knowledge repository for similar cases to the investigated one. The system afterwards retrieves the cases from the repository based on the highest similarity to the case under investigation. The similarity measures applied can vary from rather simplistic ones (if the cases have to do with numbers, symbols, and dates) to advanced ones (if the cases represent whole architectural structures or mechanical models).
2. **Reuse**: If a similar case has been found from the repository, the solution indicated inside that case will be adopted as a proposed one to the investigated case. There is high probability that the proposed solution cannot be used directly to the incoming case.

In such occasion the suggested solution has to be adapted to meet the specific requirements of the investigated case.

3. **Revise**: Since a solution has been proposed for an examined case, this solution has to be verified either in the form of a simulation or directly in a real world testing scenario. This solution often has to be revised in order to meet the needs of the investigated case accordingly.
4. **Retain**: If the above steps have been conducted successfully and a verified solution has been given to the investigated problem, then a new case can be formulated. This new case will consist of the imported problem, its proven solution plus any available surrounding information. Then this new, complete case can be stored to the knowledge repository for possible future use.

The CBR cycle can be used over a wide range of applications due to its generic characteristics and its adaptation flexibility across systems. The next section refers to the relevant work as it is being illustrated from the literature.

2.5.3 CBR Systems and Applications

Due to the generalisation of processes and the CBR-cycle portability across systems, the concept of CBR can be applied to a remarkable number of applications in a wide range of environments. This can be seen from the early days of CBR, in early 1980s, where its application included a variety of fields like those of Dynamic Memory [Schank, 1983], Analogy [Carbonell, 1983] and Legal Reasoning [Kolodner, 1983; Rissland, 1983]. This work on CBR has been continued with Hammond and Collins on case-based planning [Hammond, 1986; Hammond, 1987; Hammond, 1989; Collins, 1987] and the work of Ashley and Rissland on HYPO legal reasoning system [Ashley & Rissland, 1987].

At the same time CBR started being used in systems like the CYRUS computer system [Kolodner, 1983]. CYRUS was a question-answering system that contained information regarding the life (travels, meetings) of the former United States Secretary of State Cyrus Vance. CBR has been used in the CYRUS system to represent the memory cases that were afterwards retrieved to answer a relevant question. Based on the implementation presented in CYRUS the systems of CASEY [Koton, 1989] and MEDIATOR [Simpson, 1985] were built. CASEY, a pioneer system in the application field of diagnosis, has been used for diagnosing heart problems on patients. Its CBR mechanism was based on the existing knowledge of known heart-problem diagnoses. MEDIATOR was specialised in mediation trying to solve disputes by adapting existing dispute solutions over the new proposed ones. Michael Lebowitz has also used a CBR system to create the Integrated Partial Parser (IPP) which was "a computer system designed to read and generalize from large numbers of news stories" [Lebowitz, 1983].

Modern industrial approaches that use CBR can also be seen in the literature. An indicative example of those is the CLAVIER system. CLAVIER CBR system [Hinkle & Toomey, 1994] has been developed and used from Lockheed Corporate aircraft manufacturing company as an advisory system for their qualified personnel. The role of the system was to maximise their efficiency and ensuring the quality of composite aerospace parts before finalising them and sending them to the convection oven. Other systems like JULIA, an earlier system, has been designed to plan meals [Hinrichs, 1988; Hinrichs, 1989]; KRITIK was designing mechanical assemblies by using a combination of CBR and MBR (Model-based Reasoning) [Goel, 1989; Goel & Chandrasekaran, 1989] whereas CYCLOPS has been used for landscape design [Navinchandra, 1988]. Form Tool, [Cheetham, 2005] a decision support application based on CBR, was designed for the Plastic Industry to retrieve the specified colour formulas that meet colours requests from customers (Plastic Colour Matching tool). ShapeCBR a system designed for metal casting was created to "automate the process of creation and selection of cases to populate a CBR system for retrieval of 3D shapes to assist with the design of metal castings" [Petridis, Saeed & Knight, 2010]. ShapeCBR was using graph similarity algorithms towards the efficient retrieval of similar components from the case repository based heavily on its CBR mechanism.

Another family of applications that heavily uses CBR are the so called help-desk applications. These applications are designed to support company customers by providing relative services, starting primarily with the contact security which is frequently needed by customers. Help may also be needed regarding the issues that they face. The importance of help desk applications can be rather significant since apart from supporting customers, it can work as a monitoring tool for the company to a large extent [Marcella & Middleton, 1996].

The operational spectrum of help-desk applications can vary: starting from the detailed area of technical issues and going towards the areas of customer satisfaction, user interface experience, etc. Applications like the SMART: Support Management Automated Reasoning Technology [Acorn & Walden, 1992], the Appliance Call Centre automation at General Electric [Cheetham & Goebel, 2007] and the HOMER: CAD/CAM help desk application at DaimlerChrysler [Bergmann, 2002] were built with the focus on the customer help and support. CBR can be extremely useful in systems that require a direct answer to a number of predefined questions. Literature has already shown a number of systems using it towards that direction.

2.5.4 CBR in Events Mining

Since CBR systems intend to give recommendations, solutions to incoming cases/problems the available past experience has a leading role towards the mining of a solution. In monitoring systems the identification of a new case is based on a combination of facts. First of all the system should understand which sights constitute a new case. A following step should be to identify the specific events that establish the case. Then the CBR system should be able to understand the notion behind the events and extract the underlying pattern that could indicate a mighty problematic or an acceptable state. Since CBR structures are lazy learning systems [Feldman, Gupta & Srivastava, 2010] in the case of CBR monitoring there is a substantial need for the mining of events in order to be able:

- to understand the domain, identify faulty patterns/healthy patterns
- to proceed with the classification of new cases based on their underlying pattern.

The literature can show several examples of CBR applications which are based on events mining in a number of areas like Engineering, Financial and Military sector. PROFIT, a tool developed by Bonissone & Cheetham (1998) was using CBR to estimate residential property values for real estate transactions. The system was using CBR techniques in combination with fuzzy predicates expressing preferences in determining similarities between subject and comparable properties. PROFIT was able to estimate the actual value of an investigated property by using several hundred thousand sale records of California real estate transactions.

Varma & Roddy (1999) used event mining to create a conventional diagnostic system based on CBR for the monitoring of train locomotives. Within the context of this research GE was remotely monitoring 4000 locomotives, using a case base of fault logs to diagnose if any proactive maintenance was needed. Successful diagnoses were entered into the case base supplying the input variables and the maintenance action code as the output variable.

Varma, Aggour & Bonissone (2005) performed data gathering as part of the General Electric (GE) Global Research group among military vehicles. The purpose of the investigation was to determine whether data from peer units could be used inside a fleet in order to select the optimum vehicle for a potential military operation. The rationale behind such research was the fact that the operational life of military vehicles is characterised from long periods of inactivity and short periods of intensive usage. In such performance environment making direct usage of standard statistical techniques was not efficient. Thus the vehicle data were mined to ensure the reliability of missions by using a collective of equipment peers for a given unit.

2.5.5 CBR in the control of workflows

Apart from the clearly applied industrial sector, CBR has also been used in the area of workflows. Kim et al. (2002) have proposed a document-based workflow modelling mechanism (support system) (DWMSS) for the effective reuse of design outputs using CBR. Ong et al. (2004) have used CBR to create “a Workflow Advisor system in the context of advanced aero engine fault diagnostics” for the needs of DAME e-science Grid project. Chinthaka et al. (2009) have presented an approach to Workflow composition using CBR based on the input and output characteristics of a workflow as well as the underlying components and services that can be reused.

CBR has also been used towards the recall, reuse and effective adaptation of workflows. An example of that is the work from Minor et al. on URANOS project in chip industry [Minor et al., 2007b]. Minor et al. have used CBR along with past knowledge on workflows adaptation to support users in adapting current workflows. In order for the workflow adaptation to be established, workflows have to be represented in terms of graphs [Minor et al., 2007a]. In this way they can be comparable afterwards using structural similarity measures.

Work on the adaptation of workflows has also been conducted with the focus on the modification of their active parts via late-planning, hierarchical decomposition [Van Elst et al., 2003; Freßmann et al., 2005]. Van Elst et al. are using weakly structured workflows in organisational memory information systems (OMIS). Workflows are used in order to balance the possible conflicting goals of the task-oriented information support, the characteristics of the knowledge work and the work practice of knowledge workers. Freßmann et al. are using variables from the environmental context to provide dynamic assignment of sub-workflows within the CAKE system. Work on workflows and CBR includes research conducted by applying ad-hoc changes on an investigated workflow structure [Reichert et al., 2003; Weber et al., 2005].

The work shown in the literature has extensively dealt with the composition of workflow systems and their effective adaptation using a flexible workflow technology. Business processes depicted in the form of workflows may require changes [Minor et al., 2007b]. A reason could be the operational field of the workflow. Areas such as the medicine or the chip design industry may need alterations in their applied procedures. This could be due to a number of reasons: a need to meet an unanticipated situation, a necessary major change that has to be applied, etc. The same applies to areas where workflows are being used to represent human actions in an operating environment.

Existing work in the field shows examples where workflow engineers apply changes to parts of the structural contexture of a workflow while other parts of it are being executed. Control can be applied to the structure of a workflow while its modelling and execution can be effectively represented with existing notation of workflow patterns [Van der Aalst et al., 2003]. However, current research on workflows lacks in the area of applications where uncertainty derives due to unpredictable human actions. The monitoring and control of workflows that deal with human roles [Kapetanakis et al., 2009] should take into consideration possible adaptations and overrides that are triggered in order to meet upcoming environmental changes.

Modern business process management systems offer the possibility to managers to bypass or adapt explicitly a business process in order to deal with an operational problem or a set priority. Workflows should also be able to change in order to meet collaboration requirements in cases where different parts of an organisation have to merge their processes. The same applies between collaborating organisations where several tasks, designed for the same purpose, have to be synchronised and adapted for compatibility purposes.

2.5.6 Uncertainty in CBR

The effective monitoring faces challenges in the case of flexible and adaptable workflows. The management team of a workflow can find any information available in the form of timed event logs of actions that have been generated during its execution. Any workflow actions, communication messages as well as system generated messages can be found there. A challenge in the monitoring of workflows is that even with well-designed, well-defined workflows, the actual contextual information that affected the decisions taken can be missing. This can happen in many cases, although the system has managed to capture any provided information or even the actual event path that the system has followed.

Another factor that makes more difficult the monitoring of workflows, with interfered human roles, is the inability to capture the overall contextual information and communications behind any posterior actions. Some system actions may take place in an unofficial way (e.g. manual interventions) based on informal verbal communications or meetings among actors. These actions may not be captured from the system in the form of a logged event. The structure and orchestration of a workflow does not necessarily define exclusively the final choreography and operation of the monitored workflow. As a result in order to achieve effective monitoring of workflows the factor of uncertainty has to be taken into consideration and effectively be dealt. The case of uncertainty will be extensively discussed later on in section 3.5.

2.5.7 The need for CBR Adaptation

CBR systems are greatly relying on adaptation in order to be able to provide firm reasoning to a current state of the system. Since CBR systems relate to analogy it is essential to be able to adapt past cases in order to provide profound human-acceptable arguments. However, this states a relative challenge in complex systems since adaptation is getting more difficult.

The literature provides a number of CBR systems where adaptation is prominent. Especially over the last years the Computer Cooking Competition (CCC), present in International Conferences (International Conference for Case-Based Reasoning (ICCBR) 2009, ICCBR 2010, ICCBR 2011) have shown several systems that are keen to successive adaptation of their cases. Cooking CBR systems are using sets of ontologies along with their cases in order to be able to rectify and adjust their existing recipes to the user requirements.

An example given from the CCCs is the Taaable system [Cordier et al., 2009; Badra et al., 2009] where a whole architecture has been designed and deployed to support the successful adaptation of recipes. Taaable is backboned by a semantic wiki which works as a central module to manage all the available data and their inclusive knowledge along the system. On top of that Taaable is using opportunistic adaptation knowledge discovery in an approach to gain interactive and semiautomatic learning of adaptation knowledge triggered by user-provided feedback.

Another example is ColibriCook software [De Miguel, Plaza & Díaz-Agudo, 2008], developed for the cooking domain. ColibriCook is an ontology-based CBR system which aims to the retrieval and adaptation of cooking recipes. The system is based on the robustness of jCOLIBRI2 [Díaz-Agudo et al., 2007] CBR framework which provided a basic ontology extension [Recio-García et al., 2006]. The ontology was adapted accordingly to meet the requirements of the cooking domain. CookIIS software [Hanft, Ihle & Newo, 2009] is another example of CBR-based applications that focuses on the retrieval and adaptation of cooking recipes but using a different software engineering approach: .NET web services versus the Java ones adopted from its competitors.

Work on recipe adaptation using CBR is usually reduced to the area of textual adaptation. This is due to the fact that the systems are mainly following linguistic processing in order to be able to handle the text-based recipes. Specific work on how this is processed has been presented by Dufour-Lussier et al. (2010) indicating the various stages of the processing having as a target the optimum adaptation. Towards that direction the Formal Concept Analysis method is being used combining text mining with machine learning. As a result instead of just replacing an ingredient α with an ingredient β, the actions that had to be applied on α will be substituted with the actions that have to be applied to β.

2.5.8 Temporal concept in CBR systems

In a certain number of application fields the concept of temporal reasoning is important, since actions that take place within certain time frames may have a totally different meaning. Time can also define events in terms of importance. An example could be how a monitoring system conceptualises a 5-minute time span in a police emergency system. If this time span is the response time from a crew conducting a routine patrol it can be regarded as acceptable. However, if this time span is the response time in the middle of a critical rescue operation, it definitely means that something does not go as well as it was expected.

During the latest years there is an emerging need for temporal reasoning in CBR systems. There are mainly two trends for this [Jære, Aamodt & Skalle, 2002]: The first is that CBR systems are called to solve real, challenging problems where there is need for extensive temporal reasoning (such as medical diagnosis). The second is that CBR systems have become more interactive and transparent to users. Consequently the problems are moving away from their container, towards user-interactive assistants. In this case the sequence of actions is of relative importance.

The literature can show several examples in the area of CBR systems where the temporal concepts are being taken into consideration. In the area of diagnosis and possible prediction of adverse events before they occur, work has been conducted by Netten (1998). The focus of that work was on the incorporation of time-dependent cases and temporal reasoning while attempting to configure and establish the operational conditions for technical applications. Outcome of that was the BRIDGE project whose objective was "to improve performance of operational diagnosis systems as a means to improve safety, availability, reliability, maintainability and life cycle costs of large technical applications" [Netten, 1998].

Likhachev, Kaess & Arkin (2002) have conducted research work on the behavioural parameterisation of robots using both spatial and temporal CBR. The aim of their research was to help robots learn the optimum behaviour for autonomous navigation tasks either by having the robots under (ordinary) training or by following a mission-based training. During the latter the robots could learn while trying and making mistakes during their assigned missions. Both temporal and spatial characteristics were taken into account in this research in order to be able to match the best case for a robot within its operating environment.

In the area of oil drilling temporal sequences can indicate a number of situations where call to action may be necessary. An example could be a situation where a drill string gets stuck in the borehole, and stops the drilling process. This can be an exceedingly costly problem since drilling operation hours are of high cost and the freeing of a stuck pipe is a lengthy process. As a result a stuck pipe is one of the most costly drilling problems. Jære, Aamodt & Skalle (2002) have used a CBR system which based on temporal domain representation could advise the users where and when to stop a drilling operation. Past experience could show similar cases that led to stuck pipes. Possible explanation for the cause of that case was also given based on the general domain knowledge. The implemented system has used Allen's temporal theory which is based on intervals in order to have representation close to the way the human expert "reasons in domains where qualitative changes of parameters over time are important" [Jære, Aamodt & Skalle, 2002].

A more advanced approach to the problems faced in drilling industry has been given by [Skalle & Aamodt, 2004]. This approach has combined both CBR components and model based components within the TrollCreek architecture. Overall objective of the work was to increase the efficiency and safety of the drilling process. In order to do that, past temporal experience was used to support fault diagnosis and prediction of possible undesirable events in a domain that is characterised from "uncertainty, incompletes, and change" [Skalle & Aamodt, 2004].

Additional research on Temporal CBR (TCBR) was conducted to support dam technicians in decision making [Hassin, Norwawi & Aziz, 2006]. For water reservoirs the decision either to open or close one or more spillway gates is of critical safety. Dam-specialised technicians have often to decide which reservoir spillway gate should be opened or closed to release excess water in order to maintain a safe water level. Hassin et al. have designed a CBR engine that could support temporal data mining. The implemented prototype has incorporated CBR techniques used in a temporal data application for decision recommendation, based on historical data. Temporal hydrology data were used to evaluate the decision recommended by the prototype versus the dam expert. For this research in order to be able to capture the temporal pattern plus delays, sliding window was used as the segmentation technique.

In the area of CBR expert systems Floyd & Esfandiari (2011) have worked on extracting the temporal relationships between sensory stimuli and expert actions. Their investigation was focused on how a CBR agent system [Wooldridge, 2009] could learn from the reactions of experts to specific events by observing them over a period of time. Behaviour learning would be the optimum when the agent was able to perform like an expert on a specific input.

Navarro et al. (2012) have investigated the provision of temporal bounded reasoning in multi-agent systems that manage security in industrial environments. Their proposed approach facilitates the automatic reorganisation of tasks in combination with possible faced environmental changes. Key insight of the research is the optimisation of security tasks performed and the solution of problems with temporal constraints. A modified CBR cycle is being used for this along with ambient intelligent technologies such as GPS, Wi-Fi and HSDPA.

2.6 Conclusions

Modern systems nowadays can be represented in an effective way using business processes, following established standards and representations. A key objective of this research is to investigate whether it is possible to provide intelligent monitoring in workflow systems that due to their partial capture of workflow information and context contain a high degree of uncertainty and “fuzziness” regarding the current state of the process.

The background research presented in this chapter has shown that there is a formal definition of processes. However, it is not clear whether there should be a formal definition of the temporal representation of workflow events enabling the reporting to the process auditors and stakeholders during monitoring.

This chapter looked at the current research and standards in workflow management, business process standards and their representation. It also looked at the general temporal theories that could underpin this research. CBR can be an enabling technology for the intelligent monitoring of the workflows. This chapter investigated current research in CBR including current trends of using CBR methods for workflow management.

During the last few years there is an emerging need for the intelligent monitoring of workflows. This could be observed from the latest workshop on process-oriented case-based reasoning in ICCBR 2011 where the work of Montani et al. [Montani, Leonardi & Lo Vetere, 2011], Kapetanakis et al. [Kapetanakis et al., 2011] & Minor et al. [Minor, Bergmann & Görg, 2011] have shown the need for that.

Chapter 2 has presented the existing work in business processes, workflows, temporal logic and CBR. The following chapter will present the main enterprise system that was used as the case study for this thesis. It will also explain and get in depth to the followed methodology in order to deal with the given research questions.

Chapter 3

Towards a generic approach to manage workflows using Case-based Reasoning

3.1 Introduction

Throughout the literature review in Chapter 2 it can be firmly shown that business processes can be efficiently represented both at their design and execution stage in the efficient format of a workflow. However the need in the area of business processes for a more efficient monitoring pressurises for a more formal workflow definition. This formal definition and representation of a workflow should also allow an appropriate computational, similarity measure for the business process. The efficient monitoring of the process could be based on that.

This chapter presents what is the formulated research approach towards the efficient monitoring, diagnosis and representation of workflows. In order to retort to the stated research questions a workflow orchestrated enterprise system will be used as a case study for the undergoing research.

For the needs of its monitoring, artificial intelligence and more specifically, Case-based Reasoning (CBR) is being proposed as an appropriate technique. The reason for such suggestion is that as seen from the literature, CBR has been proven effective in areas where past knowledge is required in order to make a decision. Human experts refer to past experience in order to solve certain given problems. As a result

CBR seems a natural approach compared to the way the experts work. Process monitoring is by definition an area where past knowledge is required in order to be able to understand the current status of an investigated case.

Section 3.2 describes the selected case study system in terms of its characteristics, its actor roles, and the operations allowed within it. Business rules and operation constraints are also defined. This system was used in the majority of experiments of this thesis investigating whether workflows can be subject to intelligent monitoring.

The data of the selected system were subject to data mining and temporal analysis in order to elicit their operational information. Section 3.3 refers to the investigation of the temporal relations that regulate the investigated case study. This section also refers to the available events and sequences of events found in the datasets of the investigated system.

The usage of graphs was examined in order to be able to represent the inner-data connectivity of the system and at the same time achieve visual representation. Section 3.4 describes how graph connectivity and/or representation can be associated and used for the similarity measurement of different workflow instances.

Since the data come from a business process environment where human interaction is prominent, uncertainty derives. This uncertainty can make difficult a potential monitoring attempt of a workflow orchestrated system. Therefore section 3.5 addresses the uncertainty problem and investigates certain approaches to effectively confront it.

In order to be able to compare and contrast different workflow execution traces, the usage of adequate similarity measures among graphs was investigated. Section 3.6 describes the similarity algorithms used in this research in order to calculate the similarity measures. Finally, section 3.7 contains the conclusions of this chapter, summarises and comments on the research approach adopted in this thesis.

3.2 System used as main Case study

This thesis is investigating whether there can be intelligent monitoring, diagnosis and explanation of a business process using case-based reasoning. As the literature review has shown, there are several isolated approaches regarding the intelligent business process management. The term ‘isolated’ is being used since there is no unary approach that combines the different disciplines together in order to give an answer to the addressed research. Building on the above, the existing work is dealing with distinct perspectives of the above question. These can be:

- the representation standards of a business process
- the intelligent management of a workflow
- the temporal theories for timely-related events
- the application of CBR in the area of workflows, etc.

This thesis questions whether it is possible to unify the above disciplines and be able to apply intelligent monitoring in the area of business processes. To evaluate that, a specific approach has been formulated by selecting as a basis the workflow representation of an executed business process. Based on that, a typical formal business process was selected which could be represented in a standard BPMN format. For the needs of this research several common business process characteristics were taken into account such as:

- an undergoing business process, comprised of a number of predefined workflows

- its workflows involved a number of actors interacting and communicating securely within a workflow monitoring system [Kapetanakis et al., 2010b]
- the process involved artefacts that could be tracked and/or modified within a workflow.

The list can carry on with common found characteristics of existing workflow-orchestrated business processes.

A system that presents the above characteristics is the University of Greenwich, School of Computing and Mathematical Sciences (CMS) Exam Moderation System (EMS). The investigated system deals with the monitoring of the creation, upload, moderation, modification and approval of the exam papers that are being used during the school assessment periods. The system [Kapetanakis et al., 2009a] includes several actors (coordinators, delegated coordinators, moderators, drafters, administrators, external examiners) with different roles and responsibilities. The above mentioned roles can contain limited, restricted or unlimited permissions on the actions available in the system. Figure 3-1 shows a simplified version of the business process describing roles and available activities.

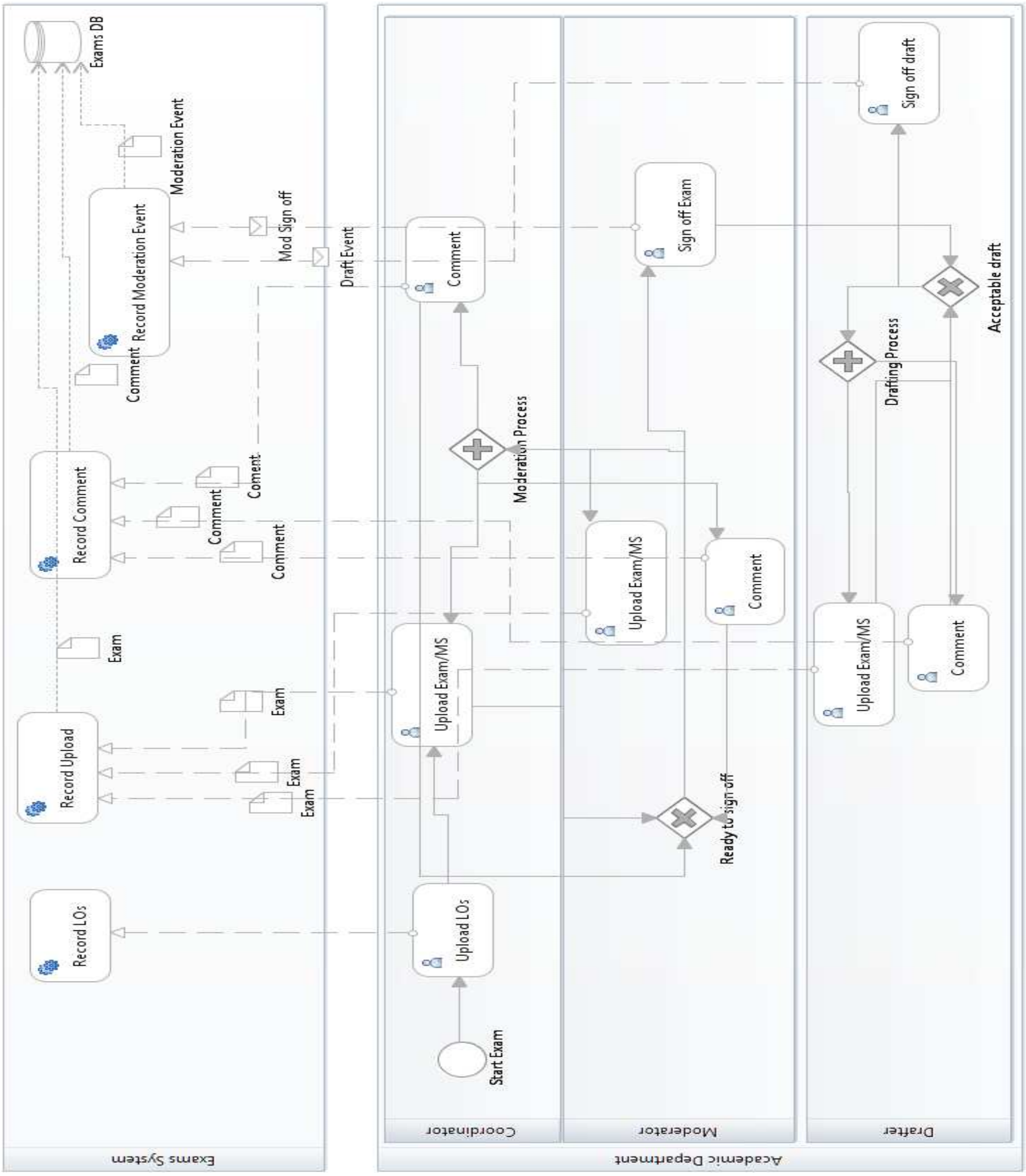

Figure 3-1: BPMN representation of the exam moderation process activities and workflows (simplified). [Kapetanakis et al, 2010a]

The main role of the system is to orchestrate the whole process and accommodate actions that deal with the different aspects of the exam creation. For any performed action the system keeps an inventory of the generated events that may refer to process stakeholders or even the system itself. Most of the workflow activities can be tracked in terms of actions, timed events and generated emails. The process overall can be defined and represented in a formal way with the help of the Unified Model Language (UML) as an activity diagram.

Several of the actions defined in the activity diagram can take place during the lifetime of an exam process. These actions can be tracked back, if necessary, since they appear into the system in the form of logged timed events. Most of the actions that take place within the system can generate a series of emails which could be distinguished in two categories: (First) emails with informative content that refer to system routine actions, (Second) targeted emails to specific workflow actors. These actors are being indicated by the workflow based on their role (they seem in charge of the next action / task).

An example of what is taking place in the system could be a new distinct upload of an examination paper by a course coordinator. The upload action, once finalised, generates a notification email to the workflow actors that the system identifies as related to the next stages of this action. While the focus of the business process is on the moderator and awaits his/her approval (See Figure 3-1), the coordinator could upload some more up-to-date versions, the administrator could change the layout of the first version and upload it as the latest one, etc.

The system contains a variety of possible events that can appear. Figure 3-2 shows the ontology of actors as it appears in the system.

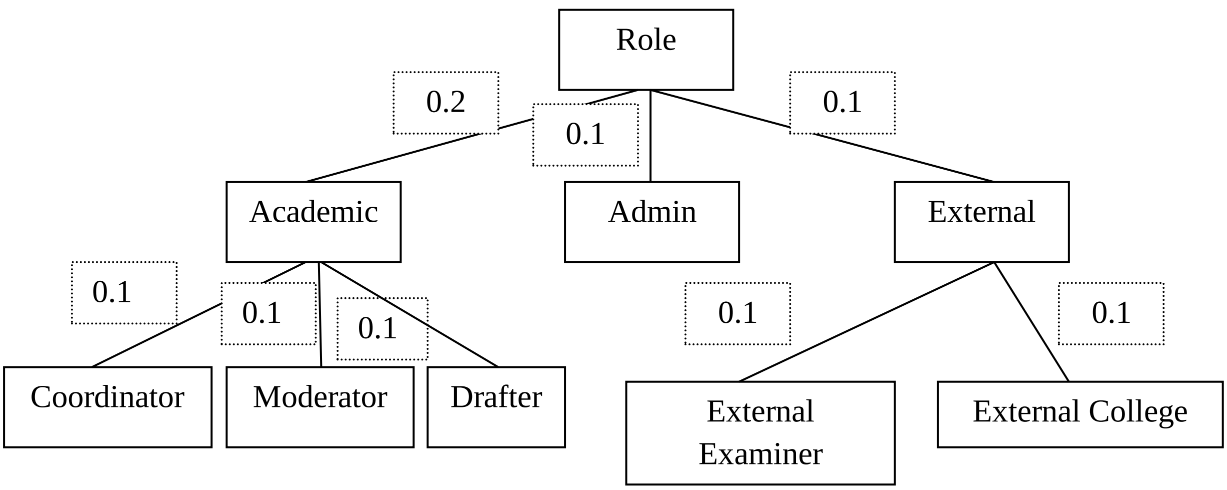

Figure 3-2: EMS Actor ontology.

Figure 3-2 shows the roles that can be found within the EMS. From the tree diagram it can be seen that several roles are close to each other. An example could be the similarities found in a number of performed tasks.

Each branch of the tree is being connected with its root with a weight indicator that can be used afterwards while calculating similarity or distance among different event sequences. The weights presented, relate to the actual distance among several roles in the system. An example follows showing how this is being conducted.

Let’s assume that we have the following actors: a Coordinator (C), a Moderator (M) and an External Examiner (EE) as indicated at figure 3-2. The actions they perform are not of interest for this example so they will be referred to as general actions (a).

Since the business process is being monitored any new action is being regarded as a case. This case is afterwards compared with past cases already present in the inventory of the monitoring system. For this indicative example the investigated case is a sequence of 2 events: Ca – Ca.

Since the investigated case is compared with existing past cases, the monitoring system recalls sequences from its inventory (case base). Its case base for this example contains just two cases: a Ca – Ma and a Ca – EEa. While the system is applying similarity measures among the investigated case and the case-base, if semantics in the form of the ontology are not taken into account, the two cases are regarded as dissimilar. However, if it takes into account the above ontology it will be keener to identify the sequence of Ca – Ma as closer to Ca – Ca compared to the Ca – EEa one.

In this way ontologies can assist significantly in the identification of patterns that are hidden at first sight due to the primitive version of the rule-set.

Figures 3-3 and 3-4 show the equivalent ontologies of artefacts and actions as they appear in the CMS Exam Moderation system.

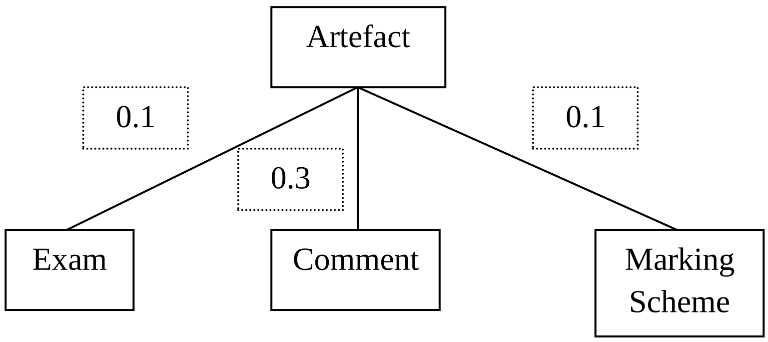

Figure 3-3: Ontology for the EMS artefacts.

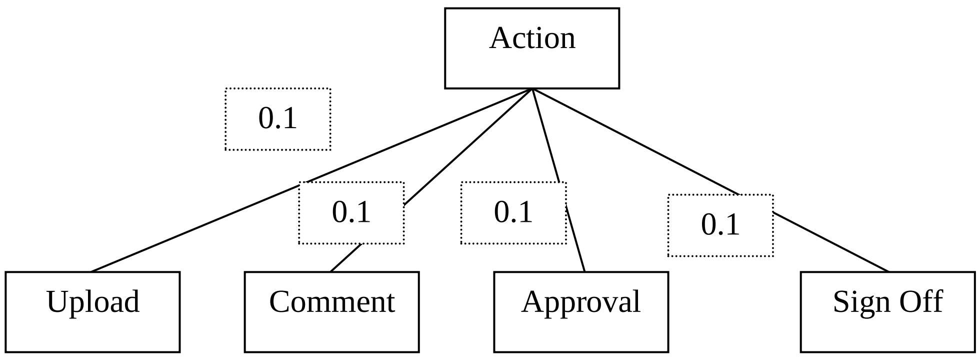

Figure 3-4: Ontology for the EMS actions.

Referring to CBR terms each case in EMS system comprises:

- a sequence of events and their temporal durations
- classification information (whether the sequence indicates a "normal" workflow execution or not)
- possible available remedial action(s)

The EMS system superintends any conducted actions, stores all uploaded artefacts as well as all the system-generated emails. At the same time it keeps proof of the conducted actions as well as any available record of the communications of actors. The system follows a formal defined business model with a plethora of pre-specified rules that work as prevention barriers to illegal or unidentified activities.

CMS Exam Moderation system does allow limited manual overrides for a number of authorised workflow actors (management team). This flexibility can put the system back on track if an unanticipated situation occurs. Although the system captures the overall bunch of actions and the internal communication, cannot predict or justify why an action has taken place and whether it is valid in the current context.

A possible drawback of the existing system is that parts of the behaviour of actors, although defined in terms of a workflow, they cannot be captured from the system. Such behaviour can contain direct emails, phone calls, face to face discussions, etc. As a result uncertainty derives related to the past or unexpected events. The notified uncertainty in the workflow results to a vague approximation of what is its current state.

The management team responsible for the monitoring of the EMS business process was interviewed in order to identify the steps required for its efficient control. The managers were able to identify potential patterns of events that could indicate either a healthy or a mighty problematic behaviour within the system. However, they were not able to characterise with absolute confidence what is the current state of the workflow. Based on the occurred events and the available communication audit trail they could presumably identify what is the current state of the workflow.

Since the business process incorporates various actors with interactive roles, there is a considerable margin for possible role overlap among them. In most cases there can be a misinterpretation of what is the current state of the workflow. Based on that it can be subsequently unclear what has to be done next.

Furthermore actors are misled in several cases by the workflow indicators. In such cases the managers should dynamically take control of the overall process in order to put it back on the right track. This can be done by either identifying what is the problem and re-assigning tasks to actors or directly initiating a series of remedial actions. The taken recovery measures could differ from manager to manager since this is subject to: firstly, what is the perception of how a problem has occurred and secondly, how the problem could be (re)solved.

The coordination of actors inside the system as well as the internal message exchange is heavily based on a targeted email generation after an action occurrence. If an email has been missed by an actor in a key position for the continuation of the process, this can be a reason for an overall process halt. In such a case no actor would interfere with the process since everybody waits for a system notification to proceed. However, such a message will never be initiated since it had been already sent (consumed).

Time intervals within the execution of the business process are a key factor in the investigation whether the process is at a healthy state or at a problematic one. Usually the entity of a time interval within the process can be interpreted as a delay between action A and action B. Certain delays are expected between actions within the system. However, it is not predefined how long these delays should be. As a result something that could be characterised as significantly slow for a certain period of the academic year (e.g. some days before an exam paper deadline), it may be acceptable during some other period of the year (e.g. the same number of days but within the period of summer holidays).

The same attitude applies to system users from a qualitative and a quantitative way affecting the processes overall. The response speed in actions can vary from actor to actor and could be fluctuating from a rather small to a rather high range. As a result there can be several slow, medium and fast responders to the system given actions. Outcome of the above observation is the following: an action which is regarded as fast for a specific actor may actually be significantly slow in terms of the response profile of a different actor. This could lead to uncertainty since it is not clear enough what a workflow auditor should expect in a number of different monitored workflow cases.

The quality of the conducted actions can also vary since the users in the system have different experience levels. As a result some actions (from specific users) may have to be repeated in order to acquire the right outcome. In order to understand the meaning of delay in an investigated case, the managers have to resort to past knowledge. This knowledge is retrieved from similar cases that took place in the past.

A possible approach to the above could be the statistical analysis of past available actions. A potential aim of this analysis could be the classification of the workflow actors according to their response rate. The application of such preliminary data analysis could help the managers formulate a view of what they should expect from the involved workflow actors and deal with them in a firm way.

3.3 Temporal Relations

The key question to address in this research is whether business processes can be monitored efficiently using Case-based Reasoning. In order to be able to monitor the execution path of a business process workflow, its occurred events as well as their exact temporal relationship has to be represented in a formal way. The reason for the strict representation formalism derives from the nature of the investigated cases. Since the majority of them come in the form of sequential timed events extracted from existing system text-logs, their formal representation ensures their exactitude, consistency and transferability.

For the needs of the temporal representation the general time theory based on intervals [Ma & Knight, 1994] was investigated. As shown in section 2.4.2.3 the general time theory extends the theory of Allen and

Hayes' (1989) by adding axioms "relating to the inclusion of time points as primitive elements" [Ma & Knight, 1994]. With this unification of points and intervals the core concepts related to temporal information are conveniently provided under a single framework. In the case of timed event logs by using Ma & Knight's (1994) time theory the temporal prepositions are reduced to the meets relationship.

The general time theory has been shown suitable for temporal similarity measure definition for CBR systems that use graph representation. This graph representation can include the events, intervals, their temporal relationships as well as similarity measures. The similarity measures can be based on graph matching techniques such as the Maximum Common Sub-graph (MCS) [Ma, Knight & Petridis, 2008; Wolf & Petridis, 2008; Petridis, Saeed & Knight, 2010]. Additionally, such a graph can be checked for consistency of temporal references using linear programming techniques [Ma, Knight & Petridis, 2008].

For example, consider a scenario with a temporal reference (T, M, D), where:

T = {t1, t2, t3, t4, t5, t6, t7, t8, t9};

M = { Meets (t1, t2), Meets (t1, t3), Meets (t2, t5),

Meets (t2, t6), Meets (t3, t4), Meets (t4, t7),

Meets (t5, t8), Meets (t6, t7), Meets (t7, t8)};

D = { Dur (t2) = 1, Dur (t4) = 0.5,

Dur (t6) = 0, Dur (t8) = 0.3}

The graphical representation of temporal reference (T, M, D) is shown in Figure 3-5:

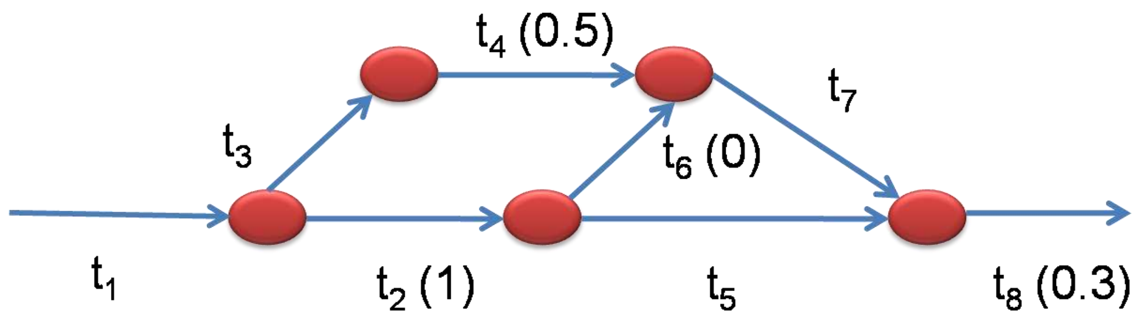

Figure 3-5: Graph representation of temporal relationships.

Workflow events are stored in an event log following a timed order. A CBR intelligent system can represent the stored events by using the general time theory based on intervals.

In the case of an investigated business process, its event logs contain sequences of events with their related timestamps. An intelligent system based on this information can highlight and classify the events with the help of the above temporal theory. However, this temporal information can be scattered. A way to collate it is needed in terms of origin and cohesion in order to be able to compare them at first stage. After their successful correlation the next task would be to be able to apply similarity measures and subsequently elicit insights. The representation of temporal information in terms of graphs could be one way towards that direction.

3.4 Graph Representation

The events taking place while a workflow is being executed are usually being logged in files using a proprietary format. While attempting to trace back an already conducted execution of a workflow, the events contained in a log can be utilised to efficiently represent the conducted operations. This information can be used afterwards for the monitoring needs from a domain expert.

While applying monitoring, the expert in charge should create a conceptual image from the sequence of the conducted events for each investigated case. Then s/he should be able to assess the case based on past experience from event sequences which are closer to the investigated one.

In the research approach presented here, the proposed intelligent system uses a similar approach to the manager's one while attempting to monitor the execution of a workflow. For each particular case study the system has access to several logs of conducted events. While analysing these logs the system is able to reproduce past event sequences based on the identified temporal knowledge. Event types are represented as nodes and the temporal characteristics of the events are used as ranked edges connecting the nodes. In this way for every sequence of events a graph is being created following similar patterns indicated at figure 3-5.

The transformation of the sequences of events into interconnected graphs changes the setting of the monitoring procedure. The similarity of objects is now reduced to similarity of graphs. Consequently the similarity calculation of objects turns into the computation of the similarity of graphs (graph matching) [Bunke, 2000].

Graphs can be convenient since they present a versatile and capable structure in the representation of objects. Graphs present some invariance properties that can be used extensively in the area of pattern extraction with traces of incidents. Bunke (2000) points out these invariance properties where "if a graph, which is drawn on paper, is translated, rotated, or transformed into its mirror image; it is still the same graph in the mathematical sense". The above properties as well as the analogy of graphs to model objects (they can preserve their relations) make them convenient for the identification of worth mentionable patterns.

By transforming the found objects into graphs and trying to apply graph matching the concepts of graph isomorphism, sub-graph isomorphism and maximum common sub-graph emerge. Graph isomorphism can be used when investigating whether two graphs are the same. Sub-graph isomorphism can be used to identify whether a part of a graph is the same with another part of the same graph or other graphs. Finally the maximum common sub-graph can be applied in any other case (where there is no isomorphism at graph or sub-graph level). A mighty obvious statement could be that "the larger the maximum common sub-graph of two graphs is, the greater is their similarity" [Bunke, 2000].

In the case of the investigated case study, the exact matching of graphs does not apply since the sequences of events can vary significantly. Another factor that has significant weight, while attempting to achieve graph matching, is the so called noise within the data. Such relevance to noise can include all the events that are not exclusively correlated within a specific trace of events. The relevance of noise within the selected case study system will be extensively discussed in section 3.6.3. Based on the above the cases of graph isomorphism and sub-graph isomorphism are deliberately neglected throughout this thesis whereas maximum common sub-graph has been broadly accepted.

An interesting approach towards the similarity calculation between two workflow graphs has been presented by Minor et al. (2007a). In their presented research approach the similarity between two graphs was not estimated based on how similar two graphs are. On the contrary, the graph distance was estimated based on how many changes one graph has to be subject to, in order to match the one in comparison. In that way the graph matching question instead of being How similar these two graphs are? is being changed to How different these two graphs are?. With this approach the distance in similarity of the 2 graphs is being measured based on how many steps are needed in order to transform the 1st graph to the 2nd one.

Regarding the addressed case study, the graph nodes are being identified by following the sequence of events identified in a log file. If the nature of events is particularly structured the produced graph follows the format indicated at figure 3-6.

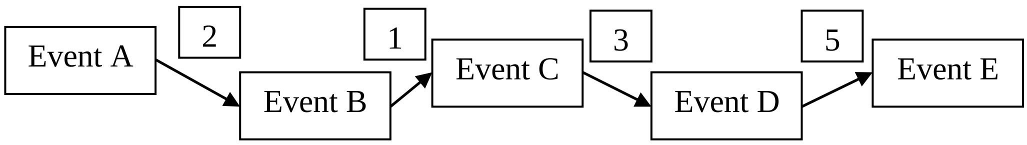

Figure 3-6: Graph representation of an event sequence.

As it can be observed from figure 3-6 if all the events and their temporal duration are known, the graph derived from the log mining can be drawn in a linear way. However, a totally different shape can be observed if several pieces of information are missing in the sequence of events. An example can be the lack of knowledge for the exact event duration or whether an event happened at an unknown temporal point. In some extreme cases there can also be that an event occurred and had never been captured by the system. Figure 3-7 shows a possible shape for the derived graph containing similar cases with the above mentioned scenario.

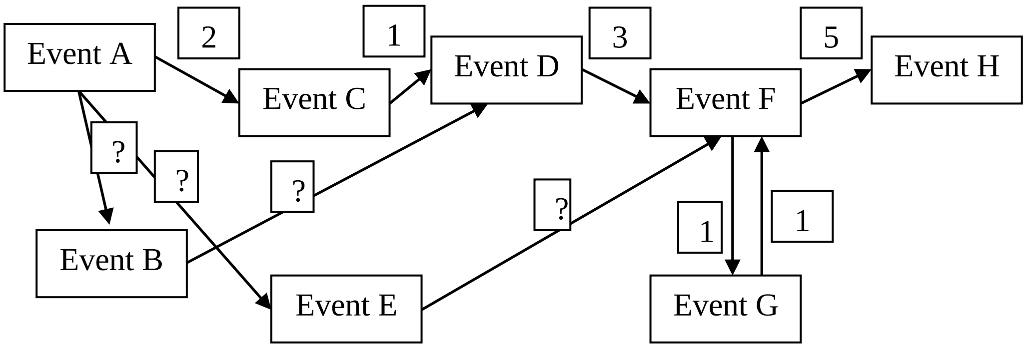

Figure 3-7: Event sequence represented as a graph. Unknown events are shown.

Graphical representations of event sequences, as shown at figure 3-7, increase the complexity while estimating the graphs similarity among graphs. However, such cases are seldom seen in the EMS case study. Cases similar to the one shown at figure 3-7 can be seen where high dimensional uncertainty is present e.g. lots of unreasoned events, unpredictable actor behaviour, weird incidents, etc. This appears often in systems where multiple actors interact with business processes in different ways and there is a large margin of flexibility in the actions that they can perform.

3.5 Reasoning under Uncertainty

An important aspect while monitoring business processes is dealing with uncertainty. Business processes deal with both humans and organisations. Humans working in organisations need to have the flexibility to work around workflows, bypassing and adapting them depending on the specific context of a contemporary condition. Such conditions could be either planned or unplanned. An example of a planned, or anticipated, condition could be the substitution of an employee in a case of an illness, or due to vacation absence. Unplanned conditions could be the so called unanticipated situations. Unanticipated situations can urge for a manual business process override. Such situations could be a natural disaster, a power outage, a committed crime within the working environment, etc.

Most workflow control systems allow humans to override and/or adapt elements of the workflow. The EMS system can usually record these changes internally. However, any context or rationale behind such changes is rarely recorded since the changes are not essentially being justified. An intelligent system needs to be able to deal with the uncertainty stemming from that. Additionally workflows can be subject to continuous change and evolution dealing with the anticipated business process needs. Similarly to the above, these changes can take place without providing a rationale or the stated context behind them. Furthermore in most of the cases, the system changes were not recorded as separate events. This is a characteristic of the EMS system that increases the uncertainty and adds difficulty to its efficient monitoring.

Any event trace from a workflow execution will only be recorded if it resides within the visibility of the workflow orchestrated system. However, a significant number of events can occur either from the interaction among people or the interaction among people and the system within the normal execution of the workflow. These events can relate to phone calls, face to face discussions, a written mail, etc., that cannot be captured from the system. As a result the recorded event information is at its best incomplete.

Most of the problems within a workflow occur due to misunderstandings of the involved actors. The misunderstandings in many cases can be regarding the current state of the workflow. They could also derive from confusion among the actors regarding their roles and/or responsibilities.

A typical example taken from the EMS system is the uncertainty within the interactions of course coordinators with the equivalent course moderators. When each actor uploads something new to the system, an automatic notification is being generated targeting the related process stakeholders. In the event of a new coordinator upload a notification is being sent to the relevant exam moderator. If this notification is being missed by the moderator then there will not be further processing. Such processing could include an exam approval, a possible new document upload or any other allowed operation from the perspective of the moderator. In such cases a deadlock will be unavoidable since: the coordinator will not take further action, the other stakeholders will wait for an action from the moderator and the moderator will remain inactive waiting for a notification from the coordinator.

A key issue contributing to the above problem is that a certain amount of delay is expected among the actions of the workflow. Therefore it might not be clear enough whether at a specific point in time:

- there is a problem within the workflow or
- a moderator is slow in processing further his/her task

Another example issue, taken from the EMS system, has to do with the limited knowledge of the system. This occurs as a result of the inability to capture the full range of contextual knowledge associated with the workflow. Communication among the system stakeholders can take place outside the system to an immeasurable extent. This could include direct emails, physical meetings, phone calls, etc. As a result several actions recorded into the system simply do not make sense from a third person’s perspective. A manager of the system should use her experience gained from the available past knowledge to rectify such situations.

As seen from the above examples uncertainty within workflow represented business processes can be due to a variety of factors. The system itself could even be another factor that causes uncertainty. Changes that take place within it are not necessarily obvious to all the related stakeholders. Therefore the possibilities for a potential misunderstanding are being increased. The reasons for that can vary due to the nature of changes, the frequency of them, the possible poor documentation, etc.

An indicative example can be extracted from the case study system. The system has been running since 2004 hosting over 500 exam workflow processes. Till the time this thesis was written it had been subject to at least 4 major alterations in its business process that were not necessarily documented. Due to this lack in the system documentation uncertainty was present in several cases where the team of experts wanted to identify a current workflow state and make a decision.

Since uncertainty is present when business processes come across with human interaction, several approaches have been suggested to deal efficiently with it. BPEL4People and Bayesian Reasoning are the most prominent ones, therefore they will be discussed in the following sections.

3.5.1 BPEL4People

Business processes can be specified appropriately with existing standards and representations as seen in section 2.2.1. However, the drawback to those standards is that human interaction is not taken into account. This can lead to a major gap while business processes are being executed. In order to fill in the above gap BPEL4People has been introduced as an innovative approach to:

- “support role based interaction of people
- assign users to generic human roles
- delegate ownership of a task to a person only
- support scenarios such as four eyes, nomination, escalation and chained execution” [IBM & SAP, 2005]

As can be seen from the above specification BPEL4People has a rather human-targeted approach. Therefore BPEL4People has been suggested as an extension of the hard-coded control systems built on top of enterprise workflow control systems. Such an approach could be suggested as a possible way to deal with vague conditions since it could enhance the reasoning within workflows regarding the activities of their actors.

However, BPEL4People is not by itself intelligent enough to address uncertainty effectively since it works as a representation standard. Intelligent monitoring systems on the other hand need to be flexible and agile enough to deal with uncertainty. A more suitable approach should be able to observe patterns of human intervention within a workflow system. This could be afterwards matched to certain conditions in the workflow and then automatically be reused as suggestion to experts with the use of a CBR system. An alternative approach to the above could be to facilitate Bayesian reasoning in combination with artificial intelligence.

3.5.2 Bayesian reasoning

Artificial monitoring of enterprise workflow systems faces the problem of dealing with partial and often uncertain information. The information available in a system may be unreliable e.g. "a patient may mis-remember when a disease started, or may not have noticed a symptom that is important to a diagnosis" [Leake, 2002]. Additionally the rules governing a system cannot take into account all the possible parameters that may make their conclusions inapplicable e.g. "the correctness of basing a diagnosis on a lab test depends whether there were conditions that might have caused a false positive, on the test being done correctly, on the results being associated with the right patient, etc." [Leake, 2002]. As a result an artificial monitoring system should be able to provide enough reasoning regarding the probability of events based on their available knowledge.

Bayesian probability can be regarded as a natural approach towards reasoning based on uncertain statements. Bayes' approach can be used to measure a state of knowledge rather than measure the frequency of explicit occurrences. The confidence of a possible suggestion can be a number within the scale of 0 and 1. The rules that a Bayesian system is using can be "justified by requirements of rationality and consistency and interpreted as an extension of logic" [Crowley, 2010]. Due to their wide applicability the basic principles of Bayesian logic have been broadly used in artificial intelligence and more explicitly in the area of machine learning.

Bayesian reasoning can be used to calculate the probabilities of events within a monitored system. Models that use Bayesian reasoning and can depict the relationships among the variables of interest in a graphical way are referred to as Bayesian networks. The literature can show examples of using Bayesian networks, applied on tasks like user modelling and medical diagnosis. An example is Intellipath, a medical system using Bayesian networks and has been incorporated in a large number of hospitals across the world for pathology diagnosis [Biocare Medical, 2009]. Medical diagnosis is a particular area which faces a great factor of uncertainty and perhaps a combination of probabilities should be taken into account. For example "a medical decision-making system might make decisions by considering the probability of a patient having a particular condition, the probability of bad side-effects of a treatment and their severity, and the probability and severity of bad effects if the treatment is not performed" [Leake, 2002].

A Bayesian approach could be used for the classification of workflow traces. From that point of view some uncertain elements can be removed and replaced by probabilistic measures that can allow a CBR system to operate in a more accurate way. An example can be the classification of users based on their response speed to workflow events in a variety of contexts. Bayes-based techniques could be used to deal with the deriving uncertainty. In such way there can be a probabilistic estimation of a possible event based on the past available observations.

An example could be the following: During the last 5 working days the Personnel Manager has replied his emails. For the 4 days s/he managed to reply to everything within 12 hours. For 1 day s/he replied all his emails within 24 hours. As a result based on Bayes' approach there is 80% probability that the manager will reply to her/his inbox emails within 12 hours and 100% that s/he will reply within 24 hours.

A Bayesian approach could be used to identify norms and outlier behaviour in the execution of workflows. However, the structural representation of workflows in terms of graphs makes the followed approach eventually more appropriate. A Bayesian approach could be used in further work, especially to deal with uncertainty issues were norms and probability of diverging behaviours could be modelled.

3.6 Similarity Measures

As seen in section 3.4 above the operational logs contain sequences of events found that can be represented in a graph format. While monitoring a business process the process manager may want to investigate how similar a new case is with those occurred in the past. In order to do this, a similarity measure is needed to estimate the similarity of the current trace of events with any available past cases. Based on the calculated similarity the underlying knowledge of the similar past traces can be used to classify the investigated case. These similarity measures along with the retrieved past knowledge can be visualised afterwards and used as explanation to the monitoring recommendations or suggestions.

Since the sequences of events can be represented and handled in terms of a graph, the algorithms used for graph-matching can be used for the calculation of the similarity measures among workflow cases.

For the purpose of graph matching several algorithms exist that cover the overall spectrum of different graphs. The algorithms that work with graph isomorphism or sub-graph isomorphism have been excluded since they deal with the holistic graph match and they don't handle efficiently the existence of noise, as referred to in section 3.4.

The current research deals with workflows whose tracked traces have to be compared in order to calculate the similarity among them. Therefore a graph similarity algorithm has been used for the calculation of their similarity measures.

In the workflow area the edit distance algorithm has been used to measure similarity [Minor et al., 2007a; Minor et al., 2007b] as well as its modified trace edit distance [Montani & Leonardi, 2010; Montani, Leonardi, & LoVetere, 2011; Montani & Leonardi, 2012]. However, the above approaches were not followed in this research for a number of reasons:

- the workflow representation in this research is based on the formal temporal theory and is highly structured in terms of graphs
- the use of MCS and Components have been shown appropriate in a number of contexts [Mileman et al., 2002; Wolf & Petridis, 2008; Petridis, Saeed & Knight, 2010], involving structural similarities. In highly structured graphs the MCS and Components approaches have shown to be appropriate
- the system experts when reasoning about the investigated workflows, in the cases used in this research, have invariably reasoned in terms of features and "motifs" that corresponded to specific conditions in the execution of the workflow. Therefore, a MCS & Components approach is a natural approach forward

Finally, this approach allows the identification of clusters of similar workflow executions that can be used for explanation. This would not be possible if the edit distance was used.

Referring to the adopted algorithms, the Component one is based on the simple count of similar event types and will be described in section 3.6.1. Further investigating the essence of temporal information among workflow instances, an implementation of the Maximum Common Sub-graph (MCS) has been considered with some relevant variants. MCS will be described extensively in section 3.6.2.

3.6.1 Graph Matching - Components

A first approach to graph similarity was to devise a simple count algorithm since previous research work has shown that this can be efficient [Petridis, Saeed & Knight, 2010] .The algorithm was evaluated using data from a number of simple experiments. The primitive algorithm, named Components, was based on the simple count of events of similar type that occur in each business process execution. The algorithm was based on a simple rationale:

Let's assume that there are two different event sequences under investigation. Since the events are of a specific type, if the number of different categories of events in the first sequence is the same with the number of different categories at the second one AND if the number of events for each category in the first sequence is equivalent with the number of events for each category in the second sequence THEN the two different event sequences have the SAME number of each and every event.

The mathematical representation of the algorithm can be seen in the equation **1** below.

$$S(G,G') = \sum_{i=1}^{no\ of\ event\ types} w_i \frac{N_i^2}{N_{total} \times N'_{total}} \quad \textbf{(Eq. 1)}$$

where N_i is the number of events of type i common to both business processes and N_{total}, N'_{total} are the total number of events in business workflow G and G' respectively.

w_i is the weighting factor for events of type i, typically $w_i = 1 /$ (number of event types). Just for the reader's reference the reason behind adopting multiplication was to separate better the deriving values.

The Components similarity algorithm seems a fine approach towards the calculation of graph similarities. However, it cannot take into consideration the concept of intervals and establish sufficient similarity calculations therefore a more advanced algorithm is needed. The following section will refer in details to the more effective Maximum Common Sub-graph.

3.6.2 Maximum Common Sub-graph

Section 3.4 has explained the reasons why graph isomorphism and sub-graph isomorphism have to be deliberately neglected in the current investigation due to the significant variances in event sequences and the existence of the so called noise. However, the Maximum Common Sub-graph (MCS) has been broadly accepted in the current research approach after proving its suitability in a number of cases.

The literature has shown several cases where MCS has been applied successfully in different domains. Mileman et al. (2002) have efficiently used MCS to measure graph similarity in 3-D shapes in the area of metal casting. Cunningham et al. (2004) have been using MCS to estimate sub-graph similarity among textual information within the context of CBR. Wolf & Petridis (2008) state another example of the MCS successful usage to estimate graph similarity measures among UML diagrams in the area of software design. Tsai et al. (2010) have used the MCS with CBR into process planning and die design processes for automotive panels reusing existing designs to develop new ones. Further research work in the area of metal casting has also been using effectively the MCS [Petridis, Saeed & Knight, 2010]. From the above research cases it can be seen that the MCS can be a suitable approach for the calculation of temporal graph similarity measures.

By launching similarity measures in areas where certain sub-graphs are important, the context of motifs [Wu, Harrigan & Cunningham, 2011] emerges. With the reference to a motif several small sequences of the same event types can be identified. These certain event sequences can define specific indication patterns that can have remarkable weight in the classification of a case. MCS can be fairly efficient in the identification of such motifs among the investigated graphs and their sub-graphs.

The Maximum Common Sub-graph similarity between two such graphs can be defined as:

$$S(G,G') = \frac{\left(\sum_{\substack{\text{matches}\\ C,C'\\ \text{in}\\ \text{MCSG}}} \sigma(C,C')\right)^2}{\text{count}(G).\text{count}(G')} \quad \textbf{(Eq. 2)}$$

where count(G) represents the number of edges in graph G and σ(C,C') is the similarity measure, 0≤ σ(C,C') ≤ 1, between two individual edges (intervals or events) C and C'.

In the case of time-stamped events, as extracted from a workflow event log, the duration of each interval can be calculated. As a result the graphs can collapse into a single timeline. In this case, the similarity measure is easier to calculate as the MCS is a common segment made up of events and intervals in a particular order in each of the compared workflow logs. In such a common graph segment each edge (event or interval) has a similarity measure to its counterpart in the other log that exceeds a specified threshold value ε. Equation **2** above can still be used to provide the overall similarity between the two workflows. Other branches of the graph could represent relevant temporal information to the operational context. This information could be used to interpret successfully a sequence of events. An example could be the proximity to a deadline (in the Exam Moderation System), or reminder communication broadcasts sent by managers outside the system (these could be normal direct emails).

The MCS presented in Equation **2** is being calculated using a greedy algorithm. When the algorithm is applied on a certain set composed by two graphs it returns "the largest connected common sub-graph based on a minimum similarity threshold between events and intervals present in both graphs" [Kapetanakis et al., 2011].

The pseudo-code of the calculation of the MCS as used for the needs of the CMS exam moderation system is being presented below:

```
MCG similarity algorithm (G, G')
        For all combinations of events and intervals C, C'
                Select Initial pair (C, C')
                return CalculateGraphSimilarity (G, G', C, C')

End for all combinations
End MCG similarity algorithm

CalculateGraphSimilarity (G, G', C, C')
        Define variable Sim, Sim1
        Sim1 = Calculate Similarity (C, C')
        If Sim1 > Threshold
                Sim = Sim + Sim1
                Show node as visited (C, C')
                Find all neighbour combinations ()
                BestNeighbour = 0
                For each neighbour (C1, C1')
                        Create remaining graphs (G1, G1')
                        Temp = CalculateGraphSimilarity(G1, G1', C1, C1')
                        If temp > BestNeighbour
                                BestNeighbour = Temp
                End for
                Sim=sim + BestNeighbour

End CalculateGraphSimilarity
```

This algorithm recursively assembles the MCSs between two workflows. The algorithm returns the sum (Σ) of the partial similarities between nodes (intervals and events) for the largest common sub-graph between the two graphs as in equation 2. This is then squared and divided by the product of the total lengths of the two graphs.

3.6.3 The concept of Event Sanitisation

While attempting to estimate the similarity among different event traces, both the above algorithms are taking into consideration all the available events. However, it has been observed that a human expert will not necessarily follow the same approach. This could be due to the fact that s/he can automatically skip events of minor importance and focus to the most important ones.

Usually several events within an event trace can be redundant since:

- they have been generated by the system for administrative purposes
- they refer to confirmations of activities
- they refer to user-repetitive actions, e.g. the user repeated the same action by mistake.

The above cases overall constitute the concept of “noise” within the dataset since they refer to information of minor importance. This noise would rather be eliminated while applying similarity measures among workflow execution traces. In this way the dataset cases could be refined and more valuable knowledge could be extracted. This subsequently leads to higher confidence for the system experts. The action of eliminating the noise among the data will be referred to this thesis as sanitising from now onwards.

Elimination of noise leads undeniably to better knowledge extraction for the system experts. However, it is hard to classify what is noise and what is not. A typically observed example follows, highlighting the dilemma created. Let’s assume two event sequences taken from a random system. The events in the system are referred to as letters A, B, C, D and E. Assuming that we have two traces: ABCDAE and ACDAE. Their similarity measure will be calculated with Components algorithm (seen in section 3.6.1), and it follows below:

$$S\ (ABCDAE,\ ACDAE) = \sum_{i=1}^{5} \frac{5^2}{6 \times 5} = 5/6 \approx= 83\%$$

In the above traces if event B is regarded as noise it has to be ignored from the similarity calculation. The calculation in that case will be:

$$S\ (A\not{B}CDAE,\ ACDAE) = \sum_{i=1}^{5} \frac{5^2}{5 \times 5} = 25/\ 25 = 100\%$$

ABCDAE can be regarded as rather similar to ACDAE following the first calculation but totally identical (!) if the second one is applied. However, several questions are raised with the above. Firstly: Is actually event B vital for the system while calculating the similarity of traces? And secondly: Which factors can define the actual importance of a particular event B?

Some examples follow in an investigation of what could actually be regarded as noise and what could not.

Using the EMS system there were observed several frequent patterns in terms of the main actions. Actions of major importance in the exam preparation cycle were those of the coordinator upload (CU) and the moderator upload (MU) among other routine events. These routine events could be an admin upload (AU), a coordinator comment (CC), a moderator comment (MC) or a drafter comment (DC).

Two predominant patterns based on the main events were those of the:

- CU MU CU which indicated that there was an exchange of important documents and the
- CU CU which indicated that the coordinator was possibly working hectically on the creation of an exam paper

Given a couple of event sequences like CU MU CU CC AU DC and CU CU CC AU DC if the MU event was misclassified as noise it would be ignored from the similarity calculations. In this particular case such an action would definitely lead to information loss since MU was a vital pattern-indicator event.

However, this may not be the case in other application domains. Such a domain could be that of online purchases by using a credit card. Let's assume that we have two types of events such as eBay and Amazon transactions, defined as (ET) and (AT) respectively. By having two sequences of transactions in the following format: ET AT ET and ET ET ET a possible similarity calculation could ignore the AT transaction since it does not indicate anything useful. A possible removal of such a transaction could be acceptable since it does not indicate anything useful and could be regarded as noise.

The application of sanitisation on a certain dataset is heavily based on the application domain area. This is due to the different importance (gravity) that its obtainable events could have. Referring to the investigated CMS exam moderation system several events could have trivial impact in the similarity measures and thus could be regarded as noise (e.g. administrative actions). The elimination of such events was necessary since they could totally mislead the graph similarity algorithms in any different case. However, misclassification of academic actions as noise could affect the system in a negative way, leading unavoidably to ambiguous results. Therefore in the followed approach, the events of the EMS were classified as important or not based on the system ontologies as presented in section 3.2. These ontologies gave to the events an appropriate weight referring to their actual importance within the system. By applying event sanitisation the graph similarity algorithms could provide more accurate results, assisting vigorously the decision support mechanism for the system experts.

3.7 Conclusions

Chapter 3 introduced the research approach towards the investigation of the research questions stated in Chapter 1. The main research question leading this thesis is whether we can intelligently monitor and diagnose business processes using case-based reasoning. As a result a real world system, coordinated via a business process has been selected as a case study.

The data extracted from workflow execution traces are usually found in sequential series of events. Therefore a need emerges: to handle them and represent them in combination with their relevant temporal information. Hence a formal theory of time has been selected and presented at this chapter.

In order to apply similarity measures among workflow event sequences an efficient graph representation has been selected. A strong argument for that has been the efficiency of graphs regarding information handling and its subsequent persistence. Graph representation allows the calculation of similarity measures although the graphs may or may not include the overall information. This is affected from the deriving uncertainty in business processes when there is intense human presence.

Two similarity algorithms are being presented in this chapter that are being used throughout the evaluation of both the research question and the subsidiary questions. Finally, the impact of noise in data has been discussed as well as the difficulty stated while trying to classify what should be regarded as noise and what should not.

Chapter 3 has presented the research approach. Chapter 4 will carry on with the experimental phase of the undergoing research, using the CMS exam moderation system as its case study.

Chapter 4

Experiments and Validation

4.1 Introduction

Chapter 3 gave an overview of what is the approach of the addressed thesis towards the efficient monitoring of on-going business processes. However, in order to apply this approach effectively two requirements have to be fulfilled: First to have a workflow that orchestrates and choreographs a business process and second to have an intelligent monitoring system that could supervise the workflow. For the needs of the above evaluation, a custom service-oriented architecture (SOA) [OASIS, 2006] was designed and developed. The name of the created platform is Case-based Reasoning Workflow Intelligent Monitoring System (CBR-WIMS). CBR-WIMS has been used as a test bed in this thesis investigating the grounds of the stated hypothesis. In order to do that a number of experiments has been designed and evaluated using the CBR-WIMS platform.

CBR-WIMS is developed to monitor in simulated "real-time" the current state of a workflow. Additionally it does not need to represent the actual business process for the purpose of the monitoring. However, a formal representation of the business process could be used to identify operations that break constraints as identified in the process. The main purpose of the monitoring is not to prove adherence to the workflow definition but to a typical execution of the process. Workflows stakeholders can follow an "acceptable" path within the workflow but there can be countless paths that can be followed, be "acceptable" and not approved for the smooth on-going of the process.

CBR-WIMS is using case-based reasoning in order to monitor the current state of a business workflow. In order to be able to bootstrap the system a small set of cases that has already been classified by the investigated system's experts (in the case of EMS senior staff members) is being used to classify the incoming, unknown cases. The maintenance of the used cases is based on the users of CBR-WIMS who by following the normal procedures they are using the cases and annotate them or write comments. However, a small number of cases is being used to classify the majority of the case base and equivalently a small number of annotations is needed.

In the occurrence of a minor change in the business process, this can be recorded in the CBR-WIMS and add the necessary weight to the new cases (in respect to the changed business process) compared to the past available ones. In the case of a major change, if the past cases are not of use and the CBR-WIMS will not take them into account to the future monitoring of the investigated system.

For the evaluation of the current research approach a number of different experiment sets were designed and conducted. The experiments followed an evolutionary approach in order to evaluate thoroughly the research methodology. Therefore the initial sets started with the application of simplified techniques on minimalistic datasets. Gradually the experiments were progressed with the application of advanced techniques on more sophisticated datasets, imposing both high complexity and significant size.

The aim of this chapter is to guide the reader through the evaluation process of the undergoing research and put emphasis on the produced results as well as the encountered outcomes from the series of undertaken experiments.

The breakdown of the chapter is as follows:

Section 4.2 gives an overview of the early conducted experiments using CBR-WIMS with the focus on the used similarity algorithms and the mining techniques rather than the nature of the dataset and its attributes. Section 4.3 discusses on the experiment sets with real data, taken from the CMS exam moderation system using the MCS and Components similarity algorithms. Section 4.4 addresses the research motivation for the investigation of a more sophisticated approach to the MCS algorithm and the experiments undertaken towards its evaluation. Section 4.5 unfolds the reasons that led to the adoption of clustering and gives an overview of the experiments conducted using the Agglomerative Hierarchical Clustering algorithm. Finally section 4.6 concludes with discussion on the results of the research experiments.

4.2 First set of Experiments – Simulated Workflows

At first stage a set of preliminary experiments was designed towards the evaluation of the proposed approach in Chapter 3. The focus was on whether CBR-WIMS is able to monitor efficiently and identify simple patterns in already known, pre-classified cases. The dataset chosen for this set of experiments was based on a simplified workflow process [Kapetanakis et al., 2009a]. The dataset was related to the chosen case-study enterprise system, the CMS Exam Moderation system. The exam moderation system stated a variety of cases that could often be characterised in a generic way. These could be referred to as stalled or not stalled having as a key point indicator whether they present a problem or not throughout their execution. This label classification can be used when the cases are not complex. For more complex cases another class has to be added but this will be explained further in section 4.3.2.

CBR-WIMS was called to engage recreating the role of a CMS exam moderation manager. Its role was to attempt classification provision to cases already familiar to the manager. In this experiment the system was challenged in terms of classification efficiency: its provided outputs should be as close as possible to the human ones. The experiment was designed with two major targets as a priority: Firstly to investigate whether it could perform successful monitoring to an elementary dataset and secondly to check the overall feasibility of the system.

Towards the preliminary targets set, the system took into account the behaviours of the domain experts. CMS exam moderation domain experts are senior members of staff that are experienced with the system as well as the underlying learning and quality procedures. Their responsibility is to ensure the smooth operation of the business process in combination with any imposed rules and procedures as well as any involved workflow actors.

For the needs of the current research several interviews have been conducted in order to elucidate the way they respond in a number of actions. To illustrate the way that the knowledge was elicited from the system experts, the used classification form can be seen in Appendix B (EMS form). The form contains the basic questions that the experts should answer regarding a particular case. The experts at first stage were asked to classify simplified workflow instances on whether they face a problem or not. Additionally they were asked to provide any available information regarding the investigated case. This information was provided in a free-text format.

CBR-WIMS to a large extent should be able to replicate their behaviour to a stated condition. The target of the first set of preliminary experiments was explicitly referring to that: to be able to rectify an investigated case based on the way a domain expert would do so.

Domain experts, while attempting to monitor the workflow execution traces, highlight certain areas of the provided information rather than the overall number of executed events. Being more precise, while monitoring an exam creation phase, the focus can be towards the end of the event trace. This is due to the fact that usually the late events contain the most up-to-date information and could indicate the current state of the workflow. Being more specific the nature of the events, e.g. uploads from key actors, as well as their containing temporal information could highly indicate what could be the state of the workflow. Therefore the system gave significant attention to those event traces, imitating the behaviour of a human expert.

Workflow experts were following similar knowledge discovery patterns while being in the process of monitoring. Several variations could be also applied depending on the size of the trace of the event (the larger it was the more possible was to contain malfunctions, inconsistencies, etc). Different characteristics of each investigated workflow case were also taken into account, e.g. the period of the year, the proximity to deadlines, important external events, etc.

Since one of the prior targets was the feasibility evaluation of the system while monitoring an elementary dataset, a simulation was proposed as the most suitable approach to that. Therefore a simulation was designed consisting of 320 simplified event logs. These logs were containing cut-down workflow execution traces and were used to produce the workflow case studies. For the needs of the simulation only exam uploads where taken into account. Moreover the last three uploads where chosen in each log event-sequence to represent the final case.

The log trace contained events in the following format:

(Action1, Actor1, Interval1, Action2, Actor2, Interval2, Action3, Actor3, Interval3)

An example of this could be given as:

(CoordUpload, John, 3, ModUpload, George, 0, CoordUpload, John, 5) with the time intervals specified as days.

A graphical representation of the format of the events in the log trace can be seen in Figure 4-1 below.

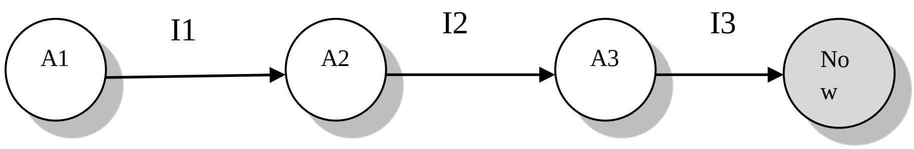

Figure 4-1: A Simple workflow Event log segment [Kapetanakis et al., 2010a].

For the needs of the efficient investigation of CBR-WIMS two aspects have been examined: first whether it can provide efficient monitoring with the actual data contained in its case base, second whether the normalisation of the behaviour of the workflow actors can assist the system in making robust judgements. Profile normalisation was considered since usually the same people tend to do same things in the same way. An example can illustrate this better:

Assuming that there is a writing process workflow (bpw) and three workflow actors: A, B and C that have to write a letter and send it within ten (10) days as part of the workflow process. After the letter is being written and sent, the action is recorded in the system. From the actors above, A writes and sends it within one (1) day, B in two (2) and C in nine (9). Without the concept of normalisation these three distinct time durations would be taken into account for any similarity calculation among the user behaviours. However, by applying normalisation there can be assumed that: the time period between one and less than three days is regarded as fast, three to five can be regarded as medium and any time period after five and less than ten days is regarded as late. By applying normalisation actors A and B can be regarded as fast and C as late. In a following way their actual time durations can be substituted with one (1) for fast, four (4) for medium and seven (7) for late or with any other indicative values for fast, medium and late responders respectively.

The normalisation perspective has been investigated during the first set of experiments with the EMS process since the actions of the workflow actors (users) and their overall behaviour (profile) could be distinguished in similar categories as above (slow, medium, fast) responders. This will be discussed in further detail in section 4.2.2.

For the following sets of experiments real data from the EMS process were used after applying an anonymisation process to the actors within the workflow executions.

4.2.1 Phase1: No normalisation on the profiles of users

During the first phase of the experiment the information elements relating to workflow actors (as individuals) were ignored. Only the actual role of the actor was taken into account.

The similarity measure in this phase contained two parts:

- Estimate the similarity between two actions (A_1, A_2)
- Estimate the similarity between the durations of two time intervals (I_1, I_2) represented as Dur(I_1) and Dur(I_2) respectively.

The similarity estimation between two actions is defined as:

- $\sigma(A_1, A_2) = 1$ if $A_1 = A_2$ and
- $\sigma(A_1, A_2) = 0$ if $A_1 \neq A_2$

The similarity measure between the durations of two intervals I_1 and I_2 is defined as:

- $\sigma(Dur(I_1), Dur(I_2)) = 1 - (|Dur(I_1)| - |Dur(I_2)|) / (|Dur(I_1)| + |Dur(I_2)|)$, $\max(|Dur(I_1)|,|Dur(I_2)|) > 0$, $\sigma(0, 0)=1$

For the estimation of the numeric relevance of two different log instances the Maximum Common Sub-graph (MCS) was used. When MCS is applied between two cases C and C', it starts from the most right (latest event) to the most left (earliest event). During its execution it calculates the similarity measures matching each interval and action in C to the corresponding one in C'. MCS stops when the similarity between two edges falls under a pre-specified threshold. The threshold placed for this experiment was at 0.5.

Just to familiarise the reader with the way the extraction of the information took place from the event logs provided, an example of a system log follows in figure 4-2. This log contains 4 distinct events.

Audit Trail
Coordinator John Smith uploaded version 2 of exam document COMP398 on Monday 19 January 2009 13:05:04
Coordinator John Smith uploaded version 3 of exam document COMP398 on Friday 23 January 2009 14:00:01
Moderator Phil Deil uploaded version 4 of exam document COMP398 on Friday 23 January 2009 15:29:21
Coordinator John Smith uploaded version 5 of exam document COMP398 on Tuesday 27 January 2009 09:05:01

Figure 4-2: An example of an audit trail log.

From figure 4-2 we can extract a random case (C). For the needs of this experiment only the last three uploads are taken into account and given another case C' we can have the following two cases:

"C= (CoordUpload, John, 3, ModUpload, Phil, 0, CoordUpload, John, 5) and

C'= (ModUpload, Phil, 4, ModUpload, Phil, 0, CoordUpload, Mary, 3)

Assembling the MCS:

1. (5, 3) = 1-2/8=0.75

2. (CoordUpload, CoordUpload) = 1

3. (0, 0) = 1

4. (ModUpload, ModUpload) = 1

5. (4, 3)=1-1/7=0.857

6. (CoordUpload, ModUpload) = 0 … MCS Matching stops

So, the overall similarity be*tween C and C' from* MCS eq. 2 as presented in section 3.6.2 is:

S (C, C') = (0.75+1+1+1+0.857)2/62=0.59" [Kapetanakis et al., 2009a]

Using the above similarity measures the sample of 320 cases was split randomly into a case base of 300 cases and a test sample (target sample) of 20 cases. The **k** nearest neighbour (kNN) algorithm with k=3 was used afterwards in order to classify the sample cases (target). Each target case was then classified as stalled or not stalled based on the votes of its three nearest neighbours. The produced results were afterwards compared against the known case classification provided by the human expert. In order to acquire more accurate results the evaluation run was repeated 10 times and the classification results were finally averaged over the 10 runs. This methodology is followed and applies to all the conducted experiments in this thesis.

The results of the evaluation runs can be found in Table 4-1.

	Average number of cases /20	**Percentage (%)**
Target cases correctly classified	13.8	69
Missed positives	5	25
False positives	1.2	6

Table 4-1: First Evaluation results – no normalisation for users' profile [Kapetanakis et al., 2009a].

As it can be seen from the above table, CBR-WIMS was able to correctly classify 69% of the investigated cases. Although the results were satisfactory for its first, pioneer usage the need to incorporate more environment information was outlined. Such information provision could possibly lead to more accurate similarity measurements.

4.2.2 Phase2: Profile Normalisation

The acquired results from the first set of experiments have shown a promising monitoring performance from CBR-WIMS. The provision of inner workflow information could possibly enhance even more the outputs of the engine. Knowledge regarding the profile of involved actors could possibly assist to further system insight provision. Motivated by the above, CBR-WIMS took into account the profiles of the human actors within the system. The results derived from the above observation have shown that the workflow actors could be divided into clusters based on their response rate. After analysing the existing cases, it was realised that they could promptly fit into one of the following three categories:

- Fast responders: response rate between 0-2 days
- Medium responders: response within 2-4 days
- Slow responders: response over 4 days

As a result, for the second phase of the experiment, normalisation was applied to the interval similarity measure. In such way the rate variations in the actor responses could be expressed more precisely among the different cases of the case base.

For the needs of this experiment the duration (I) for each interval was replaced by the difference of the actual duration minus the nominal duration for the relevant type of a workflow actor. As a result the above categories change to:

- Fast responders: response rate at 1 day
- Medium responders: response rate within 3 days
- Slow responders: response rate over 5 days

A typical example of how this alteration changes the log format can be given by changing the presented example in the beginning of section 4.2.1 above. For the needs of this example let's assume that John is a fast responder and Phil is a slow one. Such profile information could be known to a domain expert either from personal experience or from analysis of past cases where each of those actors was involved. The case could be represented in the following way based on the existing profile information:

C= (CoordUpload, John, 2, ModUpload, Phil, 5, CoordUpload, John, 4)

After applying profiling normalisation the modified similarity measure could provide additional information which was based on the knowledge extracted from the analysis of past cases. Throughout this experiment the same approach as above was followed by repeating the evaluation run for 10 times and afterwards averaging the classification results over the 10 runs. Table 4-2 summarises the produced results:

	Average number of cases / 20	**Percentage (%)**
Target cases correctly classified	15.3	76.5
Missed positives	3.8	19
False positives	0.9	4.5

Table 4-2: Second Evaluation results – normalised for actors' profiles [Kapetanakis et al., 2009a].

Phases one and two of the above experiment have shown that the correct classification within the overall number of target cases was increased after applying preliminary data profile analysis. As it can be seen from table 4-2 there was a reduction in the missed positive cases, having as a result an increase in the success rate of the classification of the cases. This has been regarded as a fairly positive monitoring attempt for the CBR-WIMS engine performing on simplistic datasets.

The first experiments with CBR-WIMS were encouraging after showing that it can effectively locate cases with either healthy or problematic condition status. These statuses were provided from workflow domain experts and were subsequently used by the system for the investigated cases.

From its primitive evaluation CBR-WIMS has shown to be effective for the monitoring of simple datasets after applying the necessary operational measures. As a further step, larger segments of data could be used. These data could be taken from a real operational cycle in order to evaluate this research approach. That was the motivation for the second set of experiments with the CBR-WIMS by using this time real workflow data taken from the EMS operational logs.

4.3 Second set of experiments – Using real workflow logs

Motivated by the encouraging results produced by the preliminary evaluation of CBR-WIMS, a second set of experiments was designed. This set took into account a complete dataset in terms of data size and its relevant complexity. The chosen dataset for this set of experiments contained real data mined from log event sequences. The chosen data were produced from the execution of the CMS exam moderation system. In this set of experiments large segments of event logs were used in order to create the case base which contained a wide range of exam moderation actions and events.

AUDIT TRAIL
Examination document Exam_COMPXX70_ver1_May_2009.DOC has been uploaded by Miltos Petridis at 03/04/2009 12:21:43. Email sent to Moderator: mcXX Office: cms-exams
Examination document Exam_MS_COMPXX70_ver1_May_2009.DOCX has been uploaded by Miltos Petridis at 04/04/2009 00:06:03. Email sent to Moderator: mcXX Office: cms-exams
Examination document Exam_COMPXX70_ver4_May_2009.DOCX has been uploaded by Ch&&&&g XX at 07/04/2009 13:15:22. Email sent to Coordinator: pmXX Office: cms-exams Head of Department: pmXX
Examination comment has been added by Ch&&&&g XX at 07/04/2009 13:18:32.

Email sent to Coordinator: pmXX Office: cms-exams Head of Department: pmXX
Moderator C&&&&g XX has signed off the Exam and Marking Scheme at 08/04/2009 00:07:17. No Emails sent
Drafter SXtX McKXXXie has removed the Moderator Sign Off for the Exam and Marking Scheme at 08/04/2009 11:40:49. Email sent to Moderator: mcXX Coordinator: pmXX Office: cms-exams Head of Department: pmXX

Figure 4-3: An example taken from an event log showing part of the exam moderation process [Kapetanakis et al., 2009b].

An example of the nature as well as the sequence of events extracted from the CMS exam moderation system logs can be seen in Figure 4-3 above. The presented events correspond to executed workflow actions. Figure 4-3 shows the system actors that triggered these workflow actions as well as the communication messages being exchanged among them.

For the needs of the evaluation of CBR-WIMS with real workflow datasets a stepwise approach was adopted. Initially familiarisation with a preliminary dataset was conducted in order to evaluate the monitoring performance of the engine. Based on the obtained results the engine came across the fully available dataset in an attempt to promote its monitoring aptitude. Such breakdown in the experiments allowed better approximation and handling of the results as well as gradual evaluation and reflection for its capabilities.

4.3.1 Work on a Preliminary dataset

As shown in section 4.2 CBR-WIMS has proven its monitoring capability in the initial attempts using simulated data. To proceed further, CBR-WIMS was challenged to classify correctly real workflow cases. As an initiation activity CBR-WIMS was trained for the classification task using the provided rules of the investigated system.

A simple experiment was designed including exam papers extracted from the CMS database. For the needs of this experiment snapshots of 29 exam papers were provided to CBR-WIMS. The exam papers were taken at the same point in time, near the exam sign off deadline. A domain expert had pre-classified those papers and had annotated 7 of them as problematic (stalled).

The role of CBR-WIMS was to classify the process states of the investigated exam papers. For the needs of the case base formulation 76 already ranked and manually classified exam moderation workflow processes were used. All the exam snapshots were chosen from the same year and their ranking (as stalled, non-stalled) had been conducted by a domain expert.

The classification of the target sample was based on simple voting using the kNN algorithm with k=3. The 3 nearest neighbours were retrieved for each case using the Maximum Common Sub-graph. Table 4-3 shows the results from the evaluation conducted.

CASES	CBR-WIMS
Correctly classified stalled workflows	**4 – 57.1%**
Missed cases	**3 – 42.9%**
False positives	**4 – 18.2%**
Correctly classified normal workflows	**18 – 81.8%**

Table 4-3: Evaluation results using real workflow logs [Kapetanakis et al., 2009a].

CBR-WIMS had a high percentage of correct ranking for normal workflow snapshots, with a percentage of 81.8%, whereas its classification efficiency for the problematic cases was 57.1%. An additional observation that came along with this experiment was that the system had additionally picked 4 more cases and identified them as stalled. When these cases where presented to a domain expert, he was able to identify 2 of those, presenting signs of potential problems that could lead to a problematic state. Compared with the previous experiment (using simulated data), this experiment did not take into account the previous behavioural knowledge for the involved actors. The case base used could also be characterised as small compared with the overall number of cases that could be available. However, these limitations were taken into consideration (and tackled) in the design of the following experiments.

This set of experiments has shown that CBR-WIMS is reliable enough to assist with the efficient classification of workflow cases. The efficient classification was proven with both simulated and real data something rather encouraging for the adopted research approach. However, this had to be further investigated with a large volume case base in order to prove its operating efficiencies.

4.3.2 Work on a Full scale dataset

CBR-WIMS has shown that it can classify with efficiency the current state of a workflow as taken at a certain point in time. However, in order to ensure the classification precision of the platform, the need for testing it with a larger and more complicated dataset emerged. This was profoundly indicated after the finalisation of the previous experiments as they were previously presented (section 4.2). Therefore the following experiments were targeted on the evaluation of the monitoring approach given a high volume of real-workflow executed traces.

Given the priorities set for this experiment the overall data-repository available from the CMS exam moderation workflow process was used. This included the total of 1588 events and 116 distinct exam moderation processes from one academic session [Kapetanakis et al., 2010a]. While incorporating the dataset into CBR–WIMS the whole range of event types was included (Uploads, Sign-offs, Comments, Reports and management actions) as well as all the available system actors (Coordinators, Moderators, Drafters and Administrators).

The data used for this experiment had significantly larger volume compared to the datasets used for the previous experiments. Therefore significant assistance had to be provided by the domain experts. A domain expert went through all the available event logs and annotated the given exam moderation processes using three distinct classes: A, B and C. Class A refers to the class stalled as mentioned in section 4.2 and class B refers to non-stalled respectively. By working closely with real data the EMS experts annotated that certain cases had problems and was rather obvious what was going on. Therefore these cases were annotated as class C.

These classes are described in more detail below:

- **Class A**: Indicates that the exam moderation process has been completed with problems. The area of problems could be a form of delay, the stall of a process at a certain time point and/or a notable confusion, discrepancy or misunderstanding among the actors of the EMS.
- **Class B**: Indicates that the exam moderation process has been completed with minor or no problems or collisions.
- **Class C**: Indicates that the exam moderation process has been stalled and was not completed correctly at the point of observation.

For this experiment a slightly different approach was adopted regarding the classification evaluation of CBR–WIMS as well as the classification measures used from the human domain expert. The reason for that was to investigate which could be the most effective configuration for the expert system in order to enhance its monitoring capabilities. For the needs of this investigation certain changes have been made in terms of the nearest neighbour configuration as well as the algorithms used.

Upon the above stated rationale, CBR–WIMS has used the kNN algorithm with k=1, 3, 5 and 7 respectively for each workflow process that was classified by the expert. An additional similarity measure algorithm was also used along with the Maximum Common Sub-graph. The second algorithm referred to as **Components** (already presented in section 3.6.1) has been based on a component event count similarity measure. In this algorithm the distance between two processes is being calculated as the normalised sum of differences between the counts of events for each type of them. This similarity measure does not take into account the comparative length of time intervals between events, but focuses on the overall number of event types involved in the process.

The number of parameters taken into consideration for this experiment was significantly larger compared to the previous performed experiments. Therefore several variants of it were conducted in order to be able to have efficient evaluation and at the same time understand the effectiveness of the current approach.

The first variants of the experiment contained all the available events triggered by the workflow actors taking into account their k (for k=1, 3, 5, 7) nearest neighbours. The key difficulty encountered with the used (large) dataset was the unavoidable interference of events with minor importance. As a result significant attention had to be given to the efficient eradication of such events from the processed event logs.

As a first stage for the log refinement event sanitisation was applied (all admin and reporting events were filtered out). Such events refer to trivial administrative operations and did not seem to have an important effect on the process. At a second filtering stage, all the actor comments were removed as well as the final actions indicated by senior managers (Drafters) of the system. This action was taken in order to focus more on the actual production stage of the exams processes. Finally a rectification algorithm, developed especially for the filtering out procedure, has been applied to smooth the event logs by consolidating similar events that happened in quick succession. The need for such algorithmic imposition emerged from the observation that actors were highly keen on repeating the same actions in a very short time span (e.g. from some seconds to one or two minutes). This could be due to an unanticipated user error (such as a wrong file upload) or due to a system error (such as a web-page refresh failure).

Figures 4-4, 4-5, 4-6, 4-7 and 4-8 below show the results of the experiments conducted as well as the percentage of correct classification for each distinct EMS class category (A, B and C) as described above.

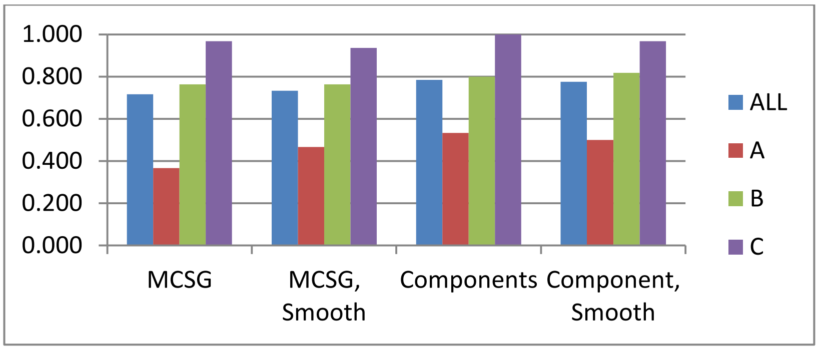

Figure 4-4: Results for 1NN and all types of events [Kapetanakis et al., 2010a].

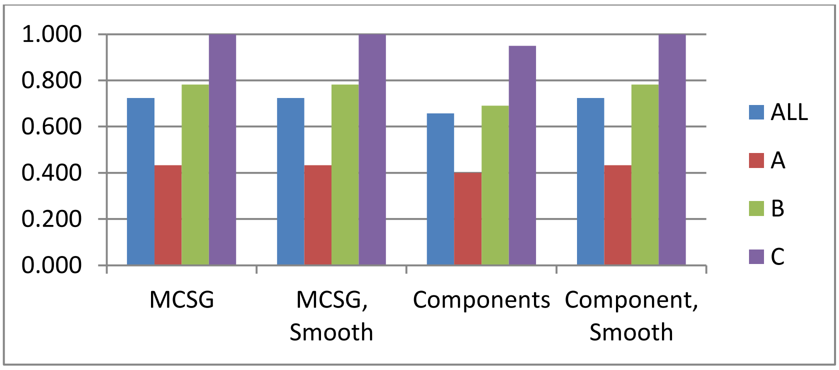

Figure 4-5: Results for 1NN and all types of events except for reports and admin actions [Kapetanakis et al., 2010a].

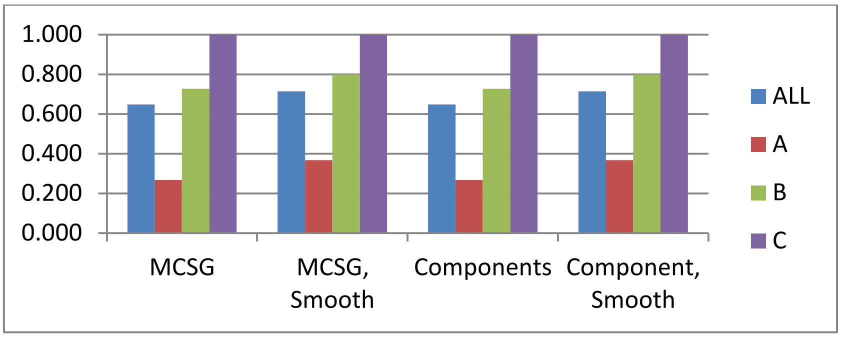

Figure 4-6: Results for 1NN and no reporting/admin or comment events [Kapetanakis et al., 2010a].

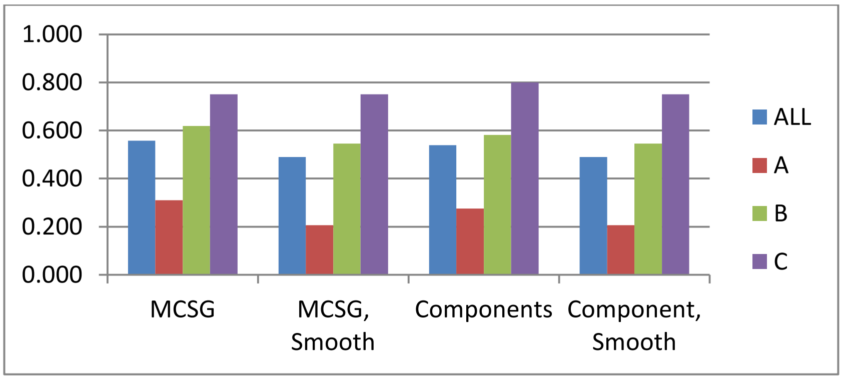

Figure 4-7: Results for 1NN only for the production stage (no drafter involvement) [Kapetanakis et al., 2010a].

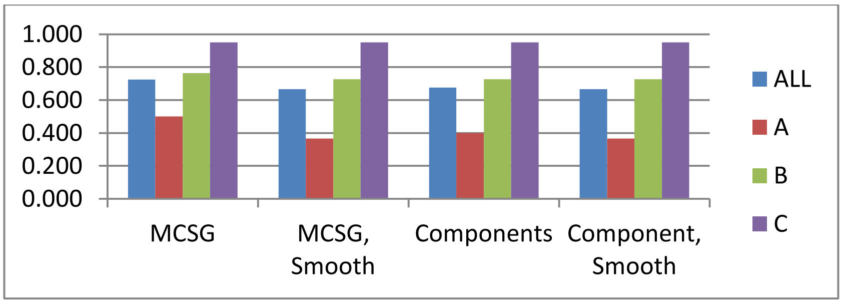

Figure 4-8: Results for 3NN no admin and reporting events [Kapetanakis et al., 2010a].

The above results can be characterised as good, especially compared to the majority-class classifier of the dataset. The majority class (B cases) gives a prediction of 55 out of 116, 47% of accuracy which is exceeded from the WIMS classification in any of the above experiments.

As it can be observed from the above stated figures CBR–WIMS was able to classify with precision the cases where severe problems occurred within the exam moderation process, mainly cases belonging to the C class. The system has been shown very accurate in terms of the classification of class B cases, the cases where the exam moderation process has been conducted in a smooth way. In cases belonging to status class A, where some problems incommoded the smooth operation of the process, CBR–WIMS has shown some reluctance in classifying them accurately. However, it was observed that the imposition of successive filters and the removal of minor importance events have improved the performance of the system. Such events included the administrator actions or the reporting events. However, the filtering process could be regarded of greater importance while applying the Maximum Common Sub-graph as the similarity measure algorithm.

This set of experiments could be regarded as the most prolific one regarding the observations and the extracted conclusions. For the needs of these experiments, CBR-WIMS has applied classification techniques using a real, complicated event-log dataset. Throughout these experiments difficulty was spotted in the identification of problems during the early stages of the process. However, that was expected since, based on the knowledge from domain experts, fewer problems occur at that early period of the academic year. The variation in the imposed kNN algorithm had also proven a factor of limited importance within the 1-3 scale and not efficient for the cases where k was 5 or 7. The latter was due to the limited number of cases within the case base.

The past behaviour norms of the human actors, in related past cases, were not taken into account for this set of experiments. Another parameter that was part of the experiment configuration was that the exam processes were investigated at a certain point in time (just before the exam-paper deadline) and not throughout a movable time window. The reason for that was the importance of this point in time for the creation of exam papers. Any other moment would be of trivial importance.

Although the acquainted results from this extensive set of experiments were positive, further questions were raised. The first question was whether CBR-WIMS efficiency was maximised or not. This could not be verified and had to be investigated further by another set of experiments. The extracted insight from the data clustering was also questionable. A raised question was whether a different clustering approach could provide different or better results.

Finally the application of a more exhaustive similarity algorithm was highlighted. Would an application of such an algorithm be able to increase the efficiency of CBR-WIMS? Following the above question, would a more analytical approach to the estimation of the similarity be of benefit? In order to find the answers to the raised questions, the idea of applying enhanced similarity measures plus a more descriptive way of clustering was formulated.

The next sections will discuss in detail both the approaches of the enhanced similarity measures and the need for a hierarchical clustering.

4.4 Enhanced Similarity Measures

The conducted experiments with CBR-WIMS platform have shown a remarkable ability towards the accurate classification of several investigated cases within the boundaries of the EMS system. However, although the results have been shown successful, a slight reservation has been underlined regarding the algorithms used for similarity measures. The arisen concern had to do with whether the used algorithms are either hiding or omitting parts of the actual case information. Those parts could possibly be of benefit to the case identification and their following system classification.

As shown in Chapter 3 the MCS is being calculated using a greedy algorithm. The algorithm returns the largest sequential common sub-graph among events and intervals present in both the investigated graphs. If the similarity measure goes beyond a pre-specified similarity threshold then the similarity calculation is being terminated. A concern raised, based on the above, is whether a more thorough calculation of the MCS, could lead to better similarity measures and therefore better monitoring insights. A possible variant of the MCS algorithm could take into consideration all the available stopping points within an investigated graph. That was the rationale for the application of the enhanced MCS within the concept of CBR-WIMS. The following section refers extensively to the proposed enhanced version of MCS as well the experiments towards its evaluation.

4.4.1 The Enhanced MCS algorithm

MCS algorithm has been proven efficient in the performed experiments, providing good prediction on the investigated cases. However, further experiments had to be conducted in order to ensure and verify its efficiency overall. Towards this direction an advanced variant of the algorithm has been developed in order to include into the similarity calculation the formerly hidden sub-graphs. These categories of sub-graphs contain small, non-overlapping sub-graphs (parts of larger graphs) that the MCS was investigating. At this stage of the experiments the research focus was on whether these graphs are of value, containing useful information for the similarity calculation.

Based on the above an advanced MCS variant has been developed. The algorithm was able to return "a set of unconnected and non-overlapping sub-graphs" [Kapetanakis et al., 2011] and could be represented formally by the equation 2 in section 3.6.2. In order to be able to hold all the available non-overlapping sub-graphs, the stopping criteria of the original algorithm had to be changed. MCS when used for calculating similarity measures it returns a value either in the case of a successful calculation or when the measure reaches a certain threshold. The developed algorithm goes beyond this stage by applying similarity measures consequently on any remaining unmatched segments of the graphs. The graphs used for this experiment were representing several workflow execution instances.

The main difference between the MCS algorithm and its advanced variant lies in the nature of the examination of the graphs. MCS requests the sub-graphs to be connected ones. However, its variant does not require that. Due to that difference the similarity measures could vary significantly compared to the ones from simple MCS.

An example illustrating how the enhanced MCS works is given below. Assuming that there are two workflow instances G and G’ with the following actions: coordinator uploads (CU), moderator uploads (MU), durations in days and coordinator comments (CC). These are workflow actions that are allowed in the EMS system. The instances follow:

G : (CU, 1.5, MU, 2, CU, 0, CC, 1, MU, 5)

G’: (CC, 3, CU, 1, MU, 2, MU, 0, MU, 0, MU, 0, MU, 0, MU, 0, CC, 1, MU, 5)

By following the MCS and having as threshold the 50% similarity, the algorithm would start from the end towards the beginning of G and G’ and would see the end sequence of “CC, 1, MU, 5” as their maximum common sub-graph. The MCS would stop when their similarity reaches the 50% threshold.

By applying the enhanced MCS, the algorithm will not stop till it calculates all the possible similarities among G and G’. The enhanced MCS would finish calculating by the time that all the unmatched segments of G and G’ are compared and their similarity is calculated. In such way it will be able to identify the “CC, 1.5, MU, 5” common sub-graph and the “CU, 1, MU, 2”. The two common sub-graph regions are highlighted below.

G : (CU, 1.5, MU, 2, CU, 0, CC, 1, MU, 5)

G’: (CC, 3, CU, 1, MU, 2, MU, 0, MU, 0, MU, 0, MU, 0, MU, 0, CC, 1, MU, 5)

The simple MCS algorithm would have picked up only the second one (towards the end) and the similarity from the second one would not contribute overall.

4.4.2 Experiments with the enhanced MCS algorithm

In order to evaluate the enhanced version of the MCS algorithm a set of experiments was designed and conducted using the CBR-WIMS platform. The focus of this set of experiments was on whether the enhanced similarity algorithm was of benefit to the predictive accuracy of the CBR process. Furthermore the computational performance was questioned: which was actually the difference in the computational overhead from the enhanced MCS algorithm compared to the original MCS one.

Towards the most accurate evaluation of the new algorithm, the earlier experiments using the CBR-WIMS [Kapetanakis et al., 2010a] were redesigned in order to include the evolved algorithm. For the needs of the evaluation 116 moderation workflow processes that included 1588 events were selected from the EMS system and were provided to the CBR-WIMS system. The predicted outcome from the CBR process was compared to the outcome (based on the A, B, C classes mentioned in section 4.3.2) that was ranked by a domain expert. The results of the investigation can be seen in Table 4-4 below.

Prediction	MCS (3NN,unfiltered)		Enhanced MCS (3NN,unfiltered)		MCS (3NN, filtered)		Enhanced MCS (3NN, filtered)	
Correct A	12	40.0%	8	26.7%	6	20%	8	26.7%
Correct B	43	78.2%	40	72.7%	43	78.2%	41	74.5%
Missed A	18	60.0%	22	73.7%	24	80%	22	73.7%
Missed B	12	21.8%	15	27.3%	12	21.8%	14	25.5%

Table 4-4: Comparison of results for the enhanced MCS similarity measures [Kapetanakis et al., 2011].

The results produced from the evaluation of the MCS variant had shown that the enhanced similarity measures did not provide greater predictive accuracy to the investigated workflow execution traces. After thorough analysis on the results of the conducted experiments, it was concluded that in most cases the connected MCS was the key indicator of a problem in the execution of a business process. The enhanced similarity measures led to the result of over-fitting. This was due to the identification of secondary patterns, common in most workflow executed traces, but not accurate predictors for problematic behaviour in the business process execution.

Samples of the achieved results have shown that the enhanced MCS can provide further discrimination among the k-nearest neighbour cases. However, the operational overhead required to calculate the similarity measures did not sufficiently justified its usage in an application context. Furthermore, it was arguable whether the produced results were driven in such way due to the nature of the application domain (and its used data) or whether they could be different (after applying the enhanced MCS to a more generalised business process).

The assessment of the enhanced version of the MCS contained a performance evaluation [Kapetanakis et al., 2011] in combination with the previous conducted experiments. The algorithm was applied on a 116 x 115 cases matrix in order to calculate the enhanced similarity measures among the cases in the case base. The overall execution time till completion was 1hr 57 seconds on a pre-set Intel Core 2 Duo

Pentium 2.16 GHz machine. The execution time on the same data matrix and using the same hardware infrastructure for MCS was 11.8 seconds. This performance indicator came on top of the previous argument that the involved computational overhead is not justified, at least for the investigated dataset. Further experiments on a different application domain may change the current research attitude but such investigation is beyond the scope of this thesis.

4.5 Hierarchical Clustering

The process of monitoring and diagnosis of a business process using intelligent techniques requires context provision and explanation to the related human stakeholders. Especially in the area of the CBR systems there is an increased need for justification since their suggested recommendations / options have to be backed-up by a stable argument. Advice and explanation provision seems essential in such cases in order to increase the confidence of the users to the proposed suggestions.

Work conducted in this area using CBR-WIMS [Kapetanakis et al., 2010b] has shown the effectiveness of a visual representation of the similarity measures applied on different business process instances. This work, stated in further detail in Chapter 6, was focused on the effective visualisation of the similarity measures in CBR-WIMS. This was also combined with the ability to drill down into individual historical business process operations. The adopted approach has been shown effective in augmenting trust to the users while proposing plausible recommendations. In such a way both the CBR and the monitoring processes rise in effectiveness.

4.5.1 The need for Hierarchical Clustering

In the research work conducted so far, the area of enhanced visualisation and the explanation of the similarity measures among cases did not provide any insight or environmental background for the retrieved cases. The extraction of contextual information was presented to the human user and reasoning was provided by presenting the retrieved, raw cases. This was an observation that urged for a different approach in reasoning. Another observation deriving from the overall application of the CBR-WIMS was the following: once a process execution trace was identified as problematic, the domain experts were able to discover specific patterns within it that indicated the existence of certain types of problems.

Based on the above observations the need for a cluster analysis emerged in order to be able to deal firmly with the stated limitations in the explanation provision of cases.

For the selection of the clustering algorithm two major factors were taken into consideration:

- The temporal complexity of the case structure
- The similarity algorithms, that provided similarity among cases instead of providing a static set of values for each case

Based on the above specification a hierarchical cluster analysis algorithm could be applied in order to identify specific types of problematic behaviours in the workflow execution. The dataset used for this experiment was taken from the EMS system. A Hierarchical Clustering (HC) algorithm was chosen to be used for the in-depth analysis of the clusters.

HC was chosen based on the nature of the provided case-base and due to its unique characteristic to organise its formed clusters in a progressive way based on their similarity distance. For the needs of the current research an Agglomerative Hierarchical Clustering algorithm was chosen in comparison to a Divisive (DHC) [Hare, 2010] one. The reason for that was that AHC follows a bottom up approach, allowing each case to formulate its own cluster. DHC follows the reverse approach (top down) treating all cases as one cluster and then splitting them while moving down the hierarchy. For the needs of this research AHC has been considered as the most suitable one since all the cases are represented individually in the case base.

Due to the distinctive bottom up characteristic of AHC, individual cases could be represented as clusters and those with the highest similarity among them could sequentially be merged into a single one. Merging continues till the pre-specified number of clusters is reached or when the clusters start becoming too diverse (this is subject to the configuration of the algorithm). A possible configuration for this could be established "by restricting the objective function which represents the mean distance of the cluster members from the notional centroid of a cluster" [Kapetanakis et al., 2011].

4.5.2 Application of AHC to CMS exam moderation system

In order to comprehensively evaluate the AHC on the provided dataset, two sets of experiments were designed. The role of the experiments was to investigate the configuration with both a fixed number of clusters and a dynamic one. The second configuration (dynamic number of defined clusters) was directed from the diversity of the cases inside the clusters. This configuration could be reduced to the centroid-based clustering [Jain, Murty & Flynn, 1999] where the cluster can be represented by a central vector. Ideally this vector states the theoretical centre-point of all the cluster cases. The application of the MCS similarity algorithm versus the Components one was also examined as part of the cluster experimentation.

AHC with MCS application

For the initial configuration of the AHC five (5) clusters were chosen as the final cluster number. MCS was afterwards applied on a combination of cases that was already ranked by a domain expert within the A, B and C distinct classes mentioned in section 4.3.2. No prior filtering was applied to the MCS cases. Tables 4-5 and 4-6 show the produced results after the application of the AHC.

Clusters	***As***	***Bs***	***Cs***	***Overall Percentage %***
Cluster 1	0	0	11	9.5
Cluster 2	1	0	13	12.1
Cluster 3	12	13	0	21.6
Cluster 4	10	22	7	33.6
Cluster 5	7	20	0	23.3

Table 4-5: Results with 5 AHC clusters based on the MCS [Kapetanakis et al., 2011].

Clusters	***As***	***Bs***	***Cs***	***Overall Percentage %***
Cluster 1	0	1	0	0.9
Cluster 2	0	0	11	9.5
Cluster 3	1	6	7	12.1
Cluster 4	1	0	13	12.1
Cluster 5	12	13	0	21.6
Cluster 6	4	7	0	9.5
Cluster 7	9	16	0	21.6
Cluster 8	3	12	0	12.9

Table 4-6: Results with 8 AHC clusters based on the MCS [Kapetanakis et al., 2011].

Experiments with both fixed and dynamic number of clusters have shown that only a few of them were key indicators of the classification of a case as a class of type **A, B** or **C**. However, there was a tendency for some clusters to group together series of events with similar behavioural characteristics. This could be something like the identification of a specific feature (e.g. the intervention of a system manager (drafter)). An example could be the roll-over of a task by sending it back to a different workflow actor (e.g. send an exam paper back for applying further changes). Such an intervention could not have necessarily led to a major problem in the operation of the business process. However, it could definitely assist when the business process becomes problematic, by providing useful context and explanation on what went wrong. Such identifications of distinct behaviours show the ability of the system to discover a variety of event patterns that can cause prospective problems.

AHC with Components Application

The first set of experiments with the AHC has used the MCS algorithm to apply similarity measures among the cases. A second set of experiments was designed, in order to further investigate whether (and if yes how) the similarity measurement algorithm affects the clustering result. The second set of experiments was based on the Components similarity measure rather than the MCS, in order to investigate the role of temporal intervals in clustering. As it has been seen from section 3.6.1 the major characteristic of the Components algorithm is its ignorance to the duration of any temporal intervals among events. Its main focus is on the number and type of events in the event-trace of the business process. Therefore it was taken into consideration for the second set of experiments.

After the application of AHC on the provided case base, the produced clusters were less able to predict the outcome as a distinct class **A, B** or **C**. However, an interesting outcome was that the clusters were able to indicate clearly the existence of attention-grabbing patterns along events. Such patterns could indicate problematic behaviours, inconsistencies or even other domain problems among the event trace.

Table 4-7 shows the results after applying AHC on the case base. Similarity measurements have been calculated using the Components similarity algorithm. Investigation on the produced results had shown that in some of the clusters (4) there was a clear prevailing event pattern, affecting all cases that were members of that cluster. Its percentage was actually the 35.4% of the whole case base. Other clusters were also containing recognisable event patterns with less obvious patterns.

Clusters	*Recognisable Event patterns*	*Pattern Details*	*As*	*Bs*	*Cs*	*Overall Percentage %*
Cluster 1			0	1	0	0.9
Cluster 2	Pattern 1	Incomplete due to delays	0	0	11	9.5
Cluster 3	Pattern 2	Incomplete due to staff problems	0	0	11	9.5
Cluster 4			1	1	0	1.7
Cluster 5	Pattern 3	Smooth operation	0	3	0	2.6
Cluster 6			1	3	0	3.4
Cluster 7			0	1	1	1.7
Cluster 8			7	16	2	21.6
Cluster 9	Pattern4	Delays in MU phase	9	3	4	13.8
Cluster 10			0	1	1	1.7
Cluster 11			5	18	1	20.7
Cluster 12			7	8	0	12.9

Table 4-7: AHC using 12 clusters with Components similarity measure and event pattern identification [Kapetanakis et al., 2011].

The results from the second experiment set, including the AHC clustering, were taken into account towards the enhancement of the explanation component of CBR-WIMS. The identified event patterns were tagged using a simple textual description, the summary of which can be seen in the table 4-7. The description was indicating the type of the pattern, its containing problem (if any) and the given advice from the experts. The advice could refer to possible ways to prevent the problem in the future or a likely remedy for that.

When a new case is being provided to CBR-WIMS for investigation, its **k**-nearest neighbour cases are being retrieved along with their known status classification: **A, B** or **C**. The system afterwards can provide a prediction to the investigated case by applying a simple voting algorithm. Furthermore the system offers the option to look at the nearest neighbours of a case. Insights of the conducted similarity measures can be provided as well as a visual representation of the applied algorithm in the form of sequential steps. Finally, the cluster tagging, referring to a possible identified event pattern, provides a more direct insight into possible case problems.

The purpose of the system was to amplify the confidence of its users regarding the extracted insights of an investigated case. Therefore the explanation component of CBR-WIMS was being used extensively. Any relevant case information, the knowledge from past cases and the calculated similarity measures were being presented and highlighted to its users in the form of visualisation techniques.

The explanation component of CBR-WIMS presents a key asset in the increase of understanding and augmented confidence while providing advice [Kapetanakis et al., 2010b]. An in depth explanation of how it works will be presented in section 6.5.

4.6 Conclusions

Chapter 4 has presented the experiments conducted with the CBR-WIMS platform towards the evaluation of the current research approach. Experiments have shown that CBR-WIMS can monitor efficiently a business process by investigating execution snapshots and applying similarity measures on them.

In the performed experiments a combination of similarity algorithms were used to enhance the classification of CBR-WIMS. Although these algorithms have been used in comparison to a large extent, finally there was no clear insight on which of them is better to use. A reason for that is that they give different answers to different questions. MCS can give an answer to qualitative questions whereas Components gives answers to quantitative questions. Components can identify a general pattern with a sort of quick look whereas MCS can give an insight where more detailed or complicated patterns exist. However, the advantages of MCS turn to be disadvantages where large quantities of data exist. A reason for that is that they can pose a significant amount of noise, as referred to in section 3.6.3.

As a conclusion it should be stated that both algorithms have been shown to be useful for the examined case study. This is because they work in a complementary mode giving a valuable overview of the actual state of the operational status among workflow execution traces.

Chapter 4 has provided an insight of how CBR-WIMS can be used for the intelligent monitoring of workflow systems. The results for the investigated workflow orchestrated system have been positive, indicating partial success of the adopted research approach. However, motivated by the main research question, the research investigation should be expanded even more since a generic approach towards workflow monitoring is pursued (thus the reference to partial success above).

Chapter 5 will address how this research approach expands over more than one workflow enacted systems as well as how the implemented software framework is suitable for this purpose.

Chapter 5

A generic architecture for the intelligent monitoring of workflows using case-based reasoning

5.1 Introduction

The previous chapter has presented the conducted experiments for the needs of the current research covering a set of different algorithms over a wide range of modulated datasets. The datasets used so far were extracted from workflow executed traces. While workflows are being executed, actions and events can be captured as well as their relevant information. The captured information can be restored afterwards and the timed path of actions can be reproduced in terms of a graph. Isolated series of events within the same path can be taken apart for investigation while monitoring the process (sub-graphs investigation).

Chapter 3 has presented the research approach and Chapter 2 has covered the current business processes standards, their workflow representation and the research work towards their efficient monitoring. However, the literature review has shown a gap in the existing technologies towards an efficient implementation that could allow intelligent workflow monitoring. Moreover in order to evaluate the presented approach in Chapter 3 and conduct successfully the experiments presented in Chapter 4 a new platform was required.

The new platform should provide two distinct characteristics: firstly comply with the existing business process standards and secondly be subject to integration in a variety of systems that contain a workflow monitoring element. The specifications of the platform were also formulated by the requirement to be integrated with complicated systems, considerably large in both numbers and range of cases. This requirement was taken into account significantly since in any other case the proposed scheme would only be able to work with elementary systems and simplified datasets.

The constitution of an effective intelligent system for the monitoring of workflows can be a puzzling task, often driven to extremes. Several challenges are stated due to the nature of workflows, their multivariate environment and the escalated complexity of their operational domains. The reason for that is that workflows are managed across enterprise systems and they should accommodate multiple interfaces. Such interfaces could provide adaptation to constantly changing systems, business processes and business needs.

Any system that can intelligently monitor workflows has to be able to provide flexibility and agility in a continuously changing environment. A suitable architecture is needed to be able to accommodate that. As a plausible consequence of the above a proposed architecture for a monitoring system has to be flexible, agile and easily adapted to a workflow managed process.

Usually such architecture should be accompanied by a resilient software framework which could provide generic functionality. Subsequently this functionality should be selectively adapted on demand, hence providing a modular software approach which at the same time is application explicit. In the case of business processes such a framework should be able to accommodate different systems with equivalently different business rules. Modern enterprise systems are advanced, that is why a proposed architecture has to be equivalently radical in order to satisfy the escalated complexity of business processes at its utmost.

In order to deal with the above stated needs of the intelligent business process monitoring, a new framework has been developed. The framework, named Case-Based Reasoning Workflow Intelligent Monitoring System (CBR-WIMS) [Kapetanakis et al., 2009a; Kapetanakis et al., 2010b; Kapetanakis et al., 2010c], consists of a series of tools and services that are encapsulated and presented under one framework to its specialised users. The aim of the development of CBR–WIMS is to provide automatic monitoring, diagnosis and explanation to workflow managers and stakeholders.

The proposed framework gives emphasis on a number of facets such as:

- the identification of potential problems within a workflow
- the analysis of workflow information
- the retrieval of past information which is similar to an investigated case
- the assembly of suggestions, recommendations that could lead to the restoration of the workflow state to an acceptable, stable (healthy) condition.

This chapter presents the whole operational spectrum behind the CBR-WIMS framework. The fundamental concepts behind its architecture are being shown including the various developed components (that deal with workflows in a generic way) and the concept of adaptors regarding its incorporated business processes. CBR-WIMS presents several generic characteristics which can be integrated with any provided business process. This chapter is also demonstrating an example of how this system could also be used across different workflow-orchestrated enterprise systems. The breakdown of the chapter is as follows:

Section 5.2 describes the architecture of CBR–WIMS, giving a contextual overview of its contained components, introduces the concept of controllers and their inner interconnections. Section 5.3

emphasises on the hierarchy of units and components within the CBR-WIMS boundaries. Section 5.4 gets in depth explaining the concept of adaptors within CBR-WIMS as well as describing their bridging characteristics between the platform and its external environment. Section 5.5 refers to the concept of CBR-WIMS and its generic attributes that lead to interoperability with other tools and platforms. This section demonstrates an example of the CBR-WIMS interoperability features having as a test bed a different workflow-oriented enterprise system. The experiments conducted throughout the evaluation of the new system are also presented at this section. Finally section 5.6 contains the conclusions of the chapter.

5.2 CBR–WIMS Architecture

Chapter 4 has addressed the effective evaluation of the research approach presented in Chapter 3. However, there are two key points that derive as a sequence of the current research approach:

- Could there be systems that can show and explain successfully, why a decision has been taken?
- Could a system be easily deployed, accessed and by using its workflow knowledge, be able to reuse this knowledge again across other systems?

In order to meet the challenges stated by the above questions a Service Oriented Architecture (SOA) has been developed and has been adopted for the conducted experiments of the current research. The developed architecture as been implemented in the form of a collection of services. This collection of services can be found under a unified tool. Its name, has been mentioned earlier (See Chapter 4), is Case-based Reasoning Workflow Intelligent Monitoring System (CBR-WIMS).

CBR–WIMS has been designed to run across a variety of business processes, therefore it contains a number of resilient components that offer high flexibility and adaptability to changes. This has been designed explicitly, in order to be able to facilitate incorporation with an imported business process.

The CBR-WIMS workflows are defined using UML activity diagrams [Kapetanakis et al., 2009a] which are mapped as Business Processes using Business Process Management Notation (BPMN) [OMG, 2011]. Their execution is being conducted via the BPEL language. For the storage needs of the workflows it is taken into account that multiple versions may be stored. A reason for that could be that a workflow is subject to changes due to the emergence of anticipated or unknown situations. For the EMS system, the exam process workflow can change across different times of the year since the requirements change (final year exam, mid-term exam, re-sit exam). Changes can also occur due to the fact that some exams may be delivered off campus. Exam papers may require external validation or collaboration with external institutions due to their delivery status.

CBR-WIMS based on the stored workflow information can dynamically orchestrate their execution in the system in an automatic way. Any workflow changes are being recorded relating to the current version of the workflow. This contributes to the creation of provenance for the different versions of the workflow and can be used as a tool to improve the user’s trust in the system. In some cases the changes in the system are followed by justifications (what was the rationale behind the change, how is related with past versions, etc.) which contribute incrementally to the system’s provenance.

CBR-WIMS was designed to allow the creation, modification and adaptation of workflows (from privileged domain users) in order to deal with the changing business process needs. The system could also allow variations in the business process execution in order to work with special business process requirements.

Business processes in CBR-WIMS are represented as workflows. The workflow descriptions are being stored in designated repositories in a process-specific temporal format based on their exclusive sequence of temporal information. Due to this temporal format the system is able to look up in past workflow executions and provide historical context based on the existing operation logs and their included events. If a business process changes, the system preserves this information, since this happens within the boundaries of the CBR-WIMS. This information can be used afterwards for post processing within the overall context of the business process.

5.3 Unit Hierarchy in CBR-WIMS

CBR-WIMS is an intelligent workflow monitoring system [Kapetanakis et al., 2009a] which incorporates a variety of components in order to conduct a wide range of operations. The system has been designed to assist in the transparent management of workflow operations. A business process should be provided as input to it and with the appropriate configuration the system is able to "orchestrate, choreograph, monitor and adapt the workflows to meet changing business processes and unanticipated operational problems and *inconsistencies*" [Kapetanakis et al., 2009a].

In order to be able to deal with business processes of scalable complexity, the system was developed with the philosophy of flexible, collaborative components. This architecture presents a similar approach to the Service Component Architecture (SCA) [OASIS, 2011] but differs in the way it approaches the design and implementation of the procedures. SCA follows the Service Oriented Architecture principles and manages to bridge together technologies with different specifications across a wide range. In this way individual components developed in different programming languages can collaborate together and combine their services besides their actual operating environment (they could even run at different operating systems). CBR-WIMS was designed and developed along with the SCA philosophy.

The overall architecture of CBR-WIMS can be seen in figure 5-1 below. Figure 5-1 presents in a diagrammatical way the collection of components and controllers that collaborate together. Possible routes of their internal message exchanges are also being indicated.

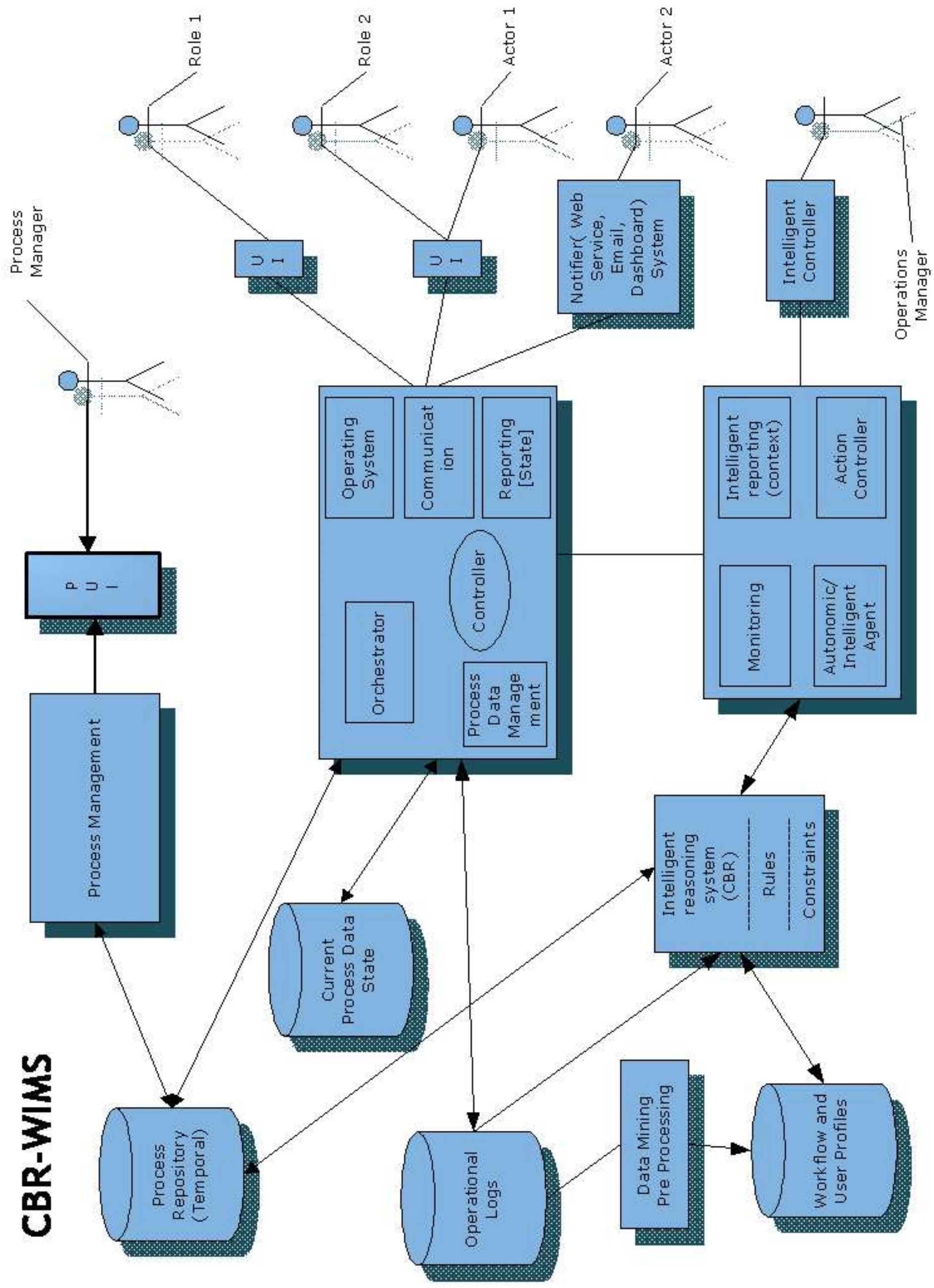

Figure 5-1: The Intelligent Workflow Management System Architecture [Kapetanakis et al., 2009a].

CBR-WIMS was developed with distinct attention to business processes and their representation since they present the core investigated entity for the system. In order to be able to coordinate the variety of operations, CBR-WIMS was internally categorised to hierarchical suites of services based on their stated functionality. These service-suites will be referred in this Chapter as Controllers, Components and Adaptors. All the aforementioned services refer both to the specialised functionalities provided by the platform as well as their complementary or combined parts.

5.3.1 Workflow Orchestrator Controller

The main part of CBR-WIMS was developed with the focus on the control of workflow operations. Since the workflows can deal with internal or external actors, the workflow controller of WIMS is also able to respond to the actions of various actors. The overall communication among them and an investigated system or deliberately among them is under the CBR-WIMS control. Additionally the system ensures that all the authorised actors are aware of events that took place at different time snapshots.

In order to achieve this, CBR-WIMS has incorporated a workflow orchestrator component that is able to consult the investigated workflow definition and orchestrate responses by delegating the appropriate Web Service(s) available. Part of the role of the orchestrator is to manage and update the stored data as well as the current state of the operation of the workflow. The control component could also provide an event audit log of the major events and actions that occurred during the workflow execution.

5.3.2 The Workflow Monitoring and Intervention Controller

In order to monitor an investigated business process, a specialised controller has been developed that is able to extract valuable information from a variety of workflow sources. The workflow monitoring and intervention controller was developed to monitor a business process and report to the workflow operation manager. Additionally the intervention controller can propose possible remedial actions based on the current state of the monitoring process.

The monitoring controller of CBR-WIMS has been developed using extensively Case-based Reasoning (CBR). The monitoring procedure is heavily based on past experience from timed event sequences as well as any related contextual information. The CBR cycle [Aamodt & Plaza, 1994] has been selected as the most efficient way of making a recommendation on an investigated case using the available past knowledge. This knowledge can be used to support a possible judgement to an investigated case based on real-time information. Such information can refer to existing environmental data, the current state of the actors, etc.

Connections with past cases are being highlighted by applying similarity measures on the investigated case across the knowledge repository (case base). Any knowledge associated with that case is also retrieved, adapted if necessary and reused formulating a possible remedial action to the investigated workflow state.

5.3.3 CBR Component

Explaining the usage of CBR in section 5.2., CBR techniques can be used in a workflow monitoring tool to extract useful precedent experience regarding known past problems. CBR-WIMS has incorporated a CBR component which is a vital part within its overall architecture. The CBR component contains a strengthened collection of fundamental methods for "the definition, orchestration and handling of the CBR process" [Kapetanakis et al., 2010c].

The CBR component can retrieve sequences of events or actions that are similar to some extent with the sequences of the investigated workflow. If a fault or a possibly problematic pattern is detected, this is reported to the operation manager along with the retrieved pattern(s) from the case base. Any associated experience that led to a possible remedy of the past case or even the solution of the past problem, are also reported to the workflow operation manager.

CBR component is a key contribution to the design and support of the Monitoring system inside CBR-WIMS since it provides the extracted past experience from any available sources. CBR component allows the classification of sequences in specific types of problems. Additionally it provides the ability to use any associate knowledge, action or experience by retrieving it, adapting and reusing it. The above abilities are taken into account when a workflow curative action is being suggested by the system.

5.3.4 Similarity Measures Component

Working along with the CBR component, the similarity calculation component is being used for the calculation of similarity measures. The applied similarity measures include the MCS and Components algorithms (See Chapter 3) as well as the advanced variant of MCS (See Chapters 3 & 4). The optimisation of MCS is explicitly configured based on the imported business process model.

The applied similarity measures are based on the graph representation of the temporal event sequences. In certain situations the similarity calculations take into account the experience extracted from past behaviours of individual workflow actors. Workflow actors can affect the overall process within a business model since their individual actions can lean the process to different boundaries. Unpredictable user actions and the rationale behind them can often lead to uncertainty.

The nature and habits of an individual user while attempting an action are being taken into account from the similarity measures component. Based on these findings the component can apply a slight weighting to the measures and normalise the calculations. This normalisation is being based on past experience and its purpose is to provide a better insight of the system while applying similarity measures. One way of retrieving the necessary past experience for the weighting purposes, is by identifying the past behaviour of actors within the workflow. For these calculations the Pre-processing component of CBR-WIMS is being used.

5.3.5 Pre–processing Component

The investigated workflow execution traces often include information that is hard to contextualise since a lot of environmental information cannot be captured by the system. This leads to uncertainty that the monitoring system is called to tackle. In order to deal with the given uncertainty and the contextual dimension of the workflow similarity, the CBR component of the WIMS engine relies on further knowledge mined from user profiles. Certain workflow norms and user behaviours that appeared in past cases are extracted by applying statistical methods and data mining pre-processing.

To assist in that a dedicated pre-processing component has been developed which is able to analyse any operational log available. The pre-processing component attempts to discover knowledge about norms and operational patterns that could be used afterwards in the calculation of the similarity measures. The work produced by the pre-processing component is of high importance to the monitoring process since it contributes to the classification of interesting behaviour or abnormal operation patterns. Classification of the above patterns cannot be attained if there is no knowledge regarding the context of what was a normal or abnormal behaviour in past. The pre-processing service contributes towards the successful retrieval of this valuable context along with the ranking provided by the system experts.

The workflow orchestrator component along with the CBR and its complementary ones, are defining a generic operational suite. This suite relates to the rules, the orchestration and the choreography of an investigated business process. Any available past knowledge along with the mined profiling of workflow actors is aggregated to the above tools leading to a robust self-manageable knowledge repository. These components give to CBR-WIMS the ability to apply workflow monitoring based on past experience. However, their functionality and specialisation is being backed up by a different concept: that of the kernel and adaptors in CBR-WIMS. These entities will be discussed in detail at the following section.

5.4 Kernel and Adaptors in CBR – WIMS

CBR-WIMS has been developed with a principal rationale: to work as a multi-discipline platform on a variety of different business processes and be able to mitigate their individual requirements. In order to deal with the given complexity from the wide range of different business processes a flexible architecture has been developed. The suggested scientific framework could support a number of different business processes that have in common the presence of uncertainty.

This section refers to the key concepts of the kernel and the adaptors in CBR-WIMS that instrument the above approach.

5.4.1 CBR-WIMS kernel

CBR-WIMS contains a collection of flexible components that can be used towards the efficient monitoring of a business process. Figure 5-1 has shown the different parts of the architectural design. In order to implement this architecture a sophisticated, core application programming interface (API) has been developed that serves as the backbone of CBR-WIMS. This core API will be referred to as the **kernel** for the rest of this thesis.

The main role of the kernel is "to identify the definition of external business process components" [Kapetanakis et al., 2010c]. Since the framework was designed with a broad vision of possible integrations with systems, its approach towards business processes is highly scalable. The kernel follows several sets of procedural identification rules in order to be able to achieve a successful ascertainment of a business process component. These sets, also regarded as modes, are initially pre-defined in the system and offer the possibility to be reorganised and/or restructured based on the individual characteristics of the investigated system.

Indicative options for the available mode-sets can be the special request mode, the learn mode and the strict mode. The special request mode allows the system to receive a new specification for a business process explicitly. Exceptions could be made for a particular input although it would not be recommended while in special request mode. The learn mode can be regarded as the best approach to a new environment where the system is always on an alert state, absorbing any new available specifications. New components are getting incorporated to the system during that mode increasing its ability to adjust in a new environment. Finally, the strict mode restricts the system to the number of business process components that the system already knows. Anything else that differs from the already acknowledged models and their rules will not be accepted by the system.

Business processes can be annotated in the form of BPMN and be stored following the XPDL format accordingly. In this way any business process actors, tasks, their inner connections, any pre – post conditions and any related constraint can be represented in a comprehended, graphical way. Since these two ways of business process representation are widely adopted, CBR-WIMS supports both of them.

CBR-WIMS can extract the information contained inside a represented model and be able to choreograph the underlying business process in terms of its actions. This could be also characterised as a sketch of a business process graph of permitted actions. The creation of such a graph for an investigated business process works as a starting point for several iterations. Each graph iteration allows the CBR-WIMS to extract a new rule for the process. In such way the system is able to generate the majority of the rules and constraints associated with a monitored case. If the rule extraction is not successful or if additional information has to be added, several alternatives exist in order to create a comprehensive set of the process constraints overall.

Every time CBR-WIMS is called to pull resources from a new business process a certain methodology is being followed in order to have successful monitoring. This methodology is characterised by a collection of steps that can be referred to as the:

- Recognition phase
- Simulation phase
- Acceptance/Rejection phase

Recognition phase

Within the recognition phase CBR-WIMS attempts to identify the behavioural needs of the imported business process structure. This can be done via a low-scale simulation of the imported system. Throughout this step the internal structure of a business process workflow is investigated and its characteristics are being indexed by the engine.

During this phase CBR-WIMS conducts a number of checks with the investigated process. This can be interpreted as a number of preliminary message-exchanges among the process and the platform. The types of messages vary and are heavily dependent on the nature of the process requests to the platform. The kernel of CBR-WIMS is in charge of estimating the necessary number of checks that is required for each investigated process. It is also related to the initiation of the message exchange, the results that derive from each process investigation and also a kind of a so called categorisation of the investigated process. By the time a new process has been successfully incorporated to the platform, CBR-WIMS can initiate direct communication with the recognised process since it has been already classified.

Simulation phase

The role of the kernel is to acquire the necessary information from an investigated business process. When the recognition phase of the business process component is finished, CBR-WIMS can highlight its internal structure. Further investigations within its inner-sections and subsections ensure that the identified structure is well understood by the engine. If these investigations are successfully completed, then the kernel classifies the business process as safe to proceed with and the different components of the platform can establish monitoring. In any other case, the kernel will try to allow partial monitoring to the known parts of the process structure.

Acceptance/Rejection phase

Ideally the kernel tries to simulate the requested actions for an investigated process before applying monitoring on both the process and its related data. If the simulation fails the engine denies applying the operation on the process as well as keeping the information into its operational logs. In case of a similar request in the future, from the same business process, the engine will refuse to accomplish it unless an expert authorises it to do so.

An equivalent policy is being applied to requests regarding the same content. The priority of CBR-WIMS is "the enforcement of the *component's data integrity and pre*-verification helps with achieving this objective" [Kapetanakis et al., 2010c]. Requests and transactions that may wobble the integrity of the investigated process are usually being rejected by the engine. The initial priority of the engine is to ensure the data integrity as well as the stability of any incorporated business process component.

5.4.2 Adaptors in CBR-WIMS

In order to be generic enough CBR-WIMS gives significant attention to the communication with internal or external factors. This can ensure its adaptability to a wide range of imported systems. The adaptors in CBR-WIMS are individual components that are being created specifically for each incorporated business process. Their main role is to adjust existing components in the engine in order to deal with the given characteristics of the investigated processes. The main component that is subject to such adaptation is the CBR component. For each imported business process, a new overlaid CBR-component is being created and customised specifically for its particular attributes.

The new overlaid component enhances the functionality of the core CBR one with information retrieved from historical data logs. These logs are being obtained from the investigated business process. History logs found in business processes are usually characterised from diversity in their format and their contextual nature. In order to be platform compliant the CBR-WIMS Parser Controller adapts the core

Parser and a new specialised one is being created, tailor-made for the investigated process.

The Parser component analyses the historical log of the given business process and renders the sequences of system events for retrieval. These events will be transformed into isolated cases that will serve as the past cases, case-base. Figure 5-2 illustrates the way in which an adaptor works in retrospect to an imported business process. As it can be seen the functionality of the WIMS Engine is being extended by an adaptor in order to meet the requirements of an imported business process.

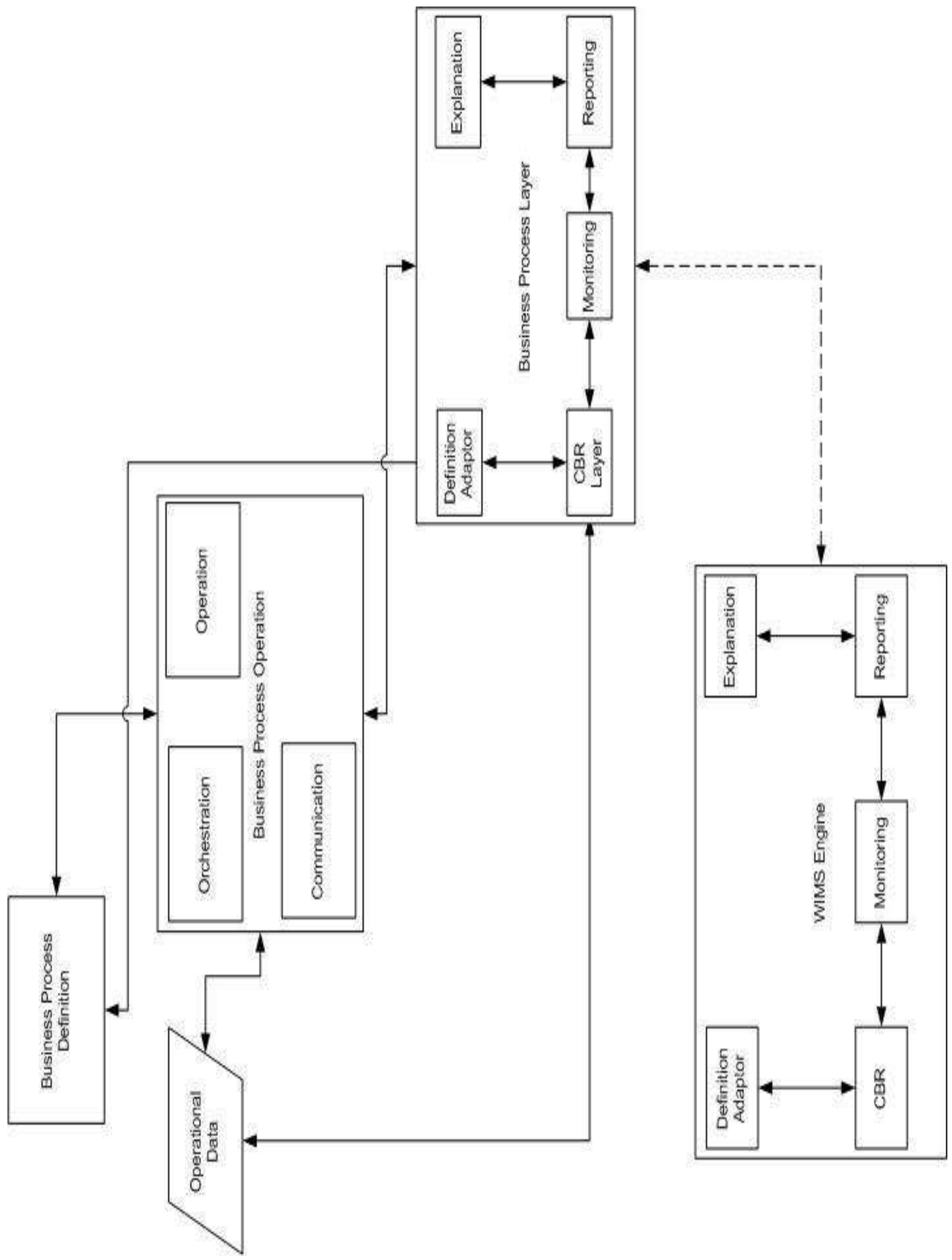

Figure 5-2: Simplified WIMS layer adapted to imported business process [Kapetanakis et al., 2010c].

Parsers in CBR-WIMS can vary since their provided functionality operates closely with their investigated systems. However, their main functionality is similar, way providing a necessary benchmark for the investigated processes. This allows the other components to work in a collaborative way within the framework.

Relating to the concept of adaptors, the CBR and Monitoring components constitute an evolutionary apparatus: since they are subject to constant changes in order to monitor effectively a new workflow execution. Referring specifically to the Monitoring component, it is constantly in an alert mode in an attempt to identify whether an observed sequence of actions is part of already known patterns. Patterns are being classified on a threat basis depending on the particular investigated system. These can be:

- Black-labelled (They caused problems to the system in the past)
- White-labelled (They pose no threat to the system)
- Grey-labelled (There cannot be safe judgment for them since either they are new or there is limited available knowledge for them)

Referring specifically to business process examples the above patterns can be seen in more than one occurrence. For example a class A workflow sequence (as described in section 4.3.2) can contain a black-labelled pattern. However, it can additionally contain a white-labelled pattern and possibly a grey-labelled one. Therefore, the class labels for cases are more abstract and refer to whole sequences of events compared to the pattern labels above, that refer to isolated collections of events within the large sequences.

The extracted information from the Monitoring component is being used by the Reporting and Explanation components of the engine in order to formulate a recommendation for the diagnosis of an investigated case. These services have to do with the similarity estimation of cases and the identification of coherence and correlation of the available workflow sequences.

The results component is the one that presents the final outcome of the system and can be used for quick reference from the workflow stakeholders. The results are accompanied by explanation that can enlighten the CBR-WIMS users and increase their confidence towards a possible decision (especially in complex cases). The Explanation component reveals how the system itself was prompted to make a certain suggestion or recommendation based on an identified situation. Explanation can also disclose how this identification was made in combination with the criteria taken into account. Any relevant information identified is also being presented, something that can differentiate if the operational domain changes.

5.5 CBR–WIMS and Interoperability

CBR-WIMS has been designed as a generic framework providing intelligent monitoring on operational workflows. Chapter 5 has presented so far the motivation behind its sophisticated architecture. The adopted architecture was moulded on the need for an efficient monitoring workflow platform. Such a platform should deal effectively with the complexity of real business processes as well as allowing knowledge-reuse across similar workflows. For the needs of the evaluation of CBR-WIMS, a real business process was used in combination with its collected data over a series of years. However, since CBR-WIMS presents a **generic** framework its monitoring abilities across different systems should be evaluated.

Motivated by the above, a different system was required for the evaluation of the generic workflow monitoring abilities of CBR-WIMS. The proposed system had to be workflow orientated, choreographed and orchestrated in order to meet the foundations of the proposed research. Section 5.5.1 presents the system chosen for the needs of the evaluation.

5.5.1 Box Tracking System

For the needs of the interoperability evaluation of CBR-WIMS an industrial business process was selected as a case study. The selected business process is an on-line archive management system named Box Tracking System (BTS). BTS is used by a local (London-based) company which operates within the area of box archiving. Its operations deal mainly with the pick-up of boxes from its clients, their storage in one of their warehouses and their return to their clients after a certain amount of time. Each available operation can be broken down to further tasks. An example can be the Box picking operation which contains the following tasks:

- Print Picking List
- Picking task
- Move to Warehouse
- Update the Picking List
- Record Investigation "Not found"
- Confirm Pricking

BTS is an automated workflow orchestrated system that deals with the efficient transfer and archiving of dedicated boxes among the clients of the company. BTS has a number of control services that the clients can use to:

- request a pickup of a box,
- authorise a transaction,
- arrange the pickup details,
- arrange the delivery details
- arrange the archiving of the box, etc.

The above can be an example of a possible sequence of transactions within the system. The reverse sequence is also possible e.g. to request a box being archived, authorise the transaction, etc. BTS allows secure communication among its actors who can be either BTS management staff or vendors.

The system can allow a large number of transactions within a very short time span. The number of archived boxes can also be of significant size imposing no limitations to its vendors. All transactions that have been authorised can be executed in combination with company vehicles that are responsible for the carriage throughout their executed routes. BTS is responsible to provide feedback at any actual state of the box throughout its lifetime. Regarding the archived items, the management team of BTS are responsible for them while residing at designated warehouses.

The BTS services are heavily dependent on the boxes since this is the basic, thus foundational unit of the process. The most important box operations are those that are connected with the actual journey of a box. These operations can take a significantly small amount of time within the lifecycle of a box but seem to be of utmost importance. Rich pictures 5.3 and 5.4 show what could be the operations Before a box*'s journey* and After a box*'s journey* respectively.

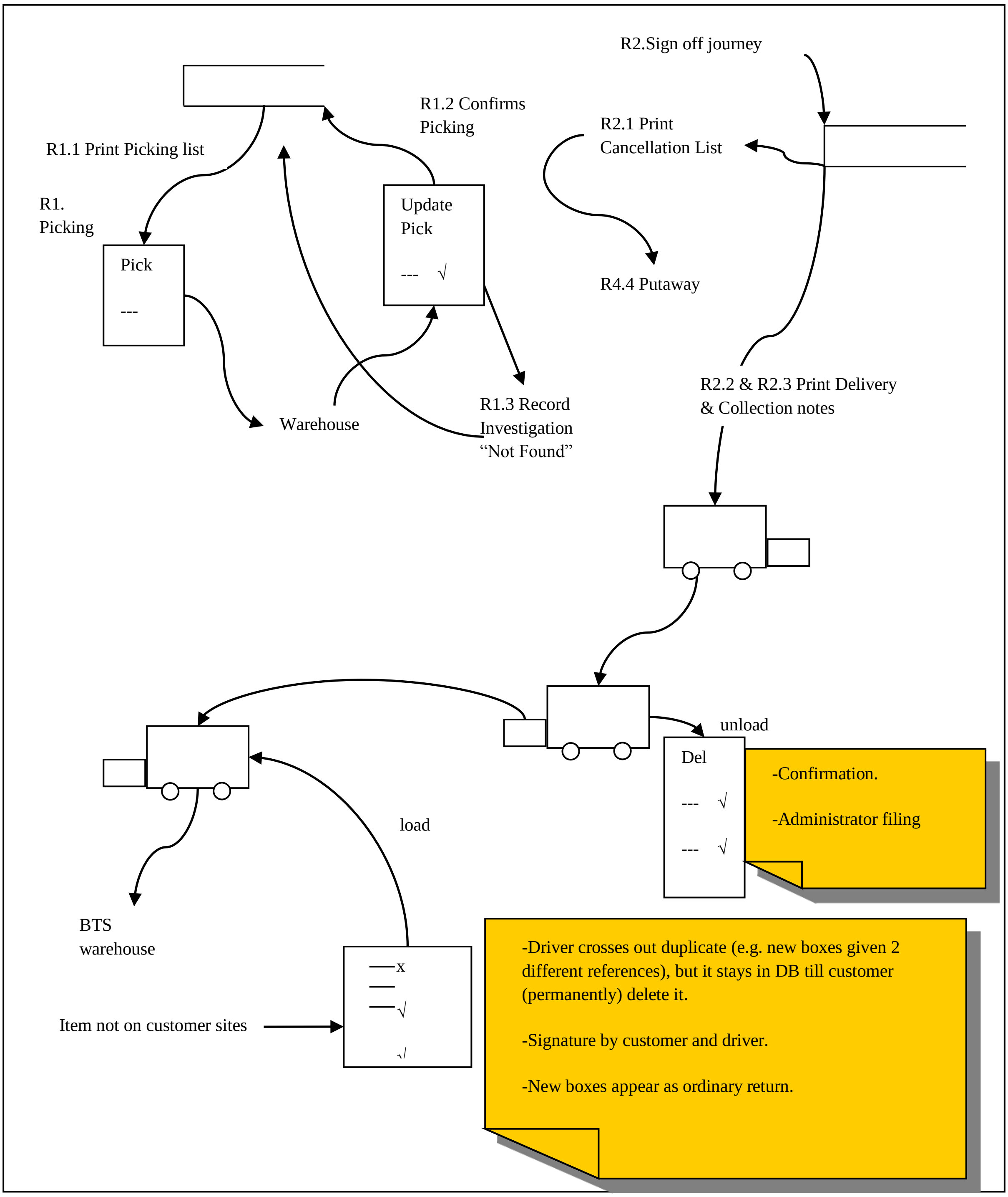

Figure 5-3: BTS - Before a box's journey.

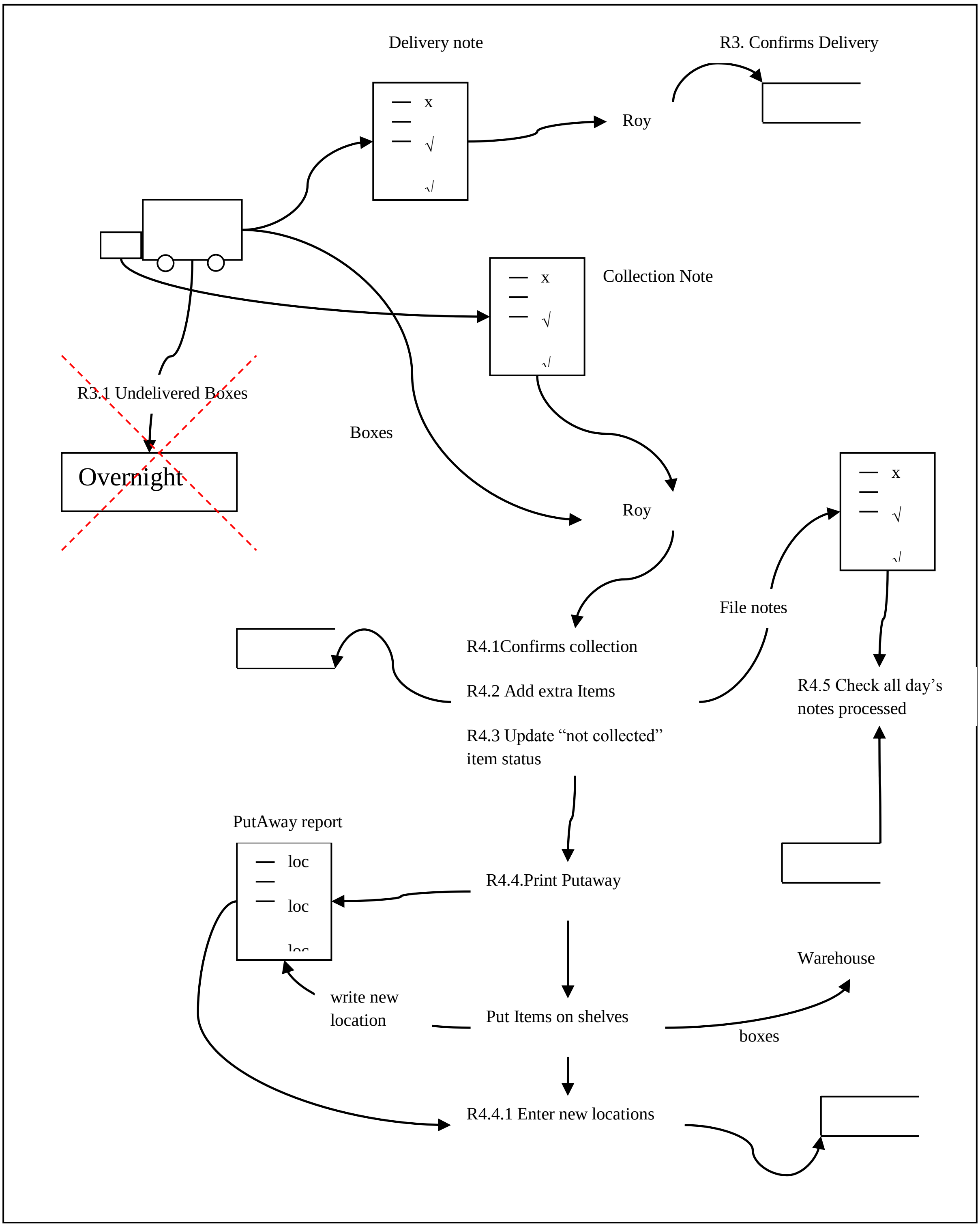

Figure 5-4: BTS - After a box's journey.

Archived units in the BTS system can remain inactive for a long period of time. This inert period can vary from several months till several years. Since boxes remain inactive they may be subject to internal changes e.g. have to be reallocated within their storage warehouse. There can also be a scenario of relocating them from warehouse A to warehouse B, from B back to A or a different one, etc. The system is able to capture partially this information since only major internal transactions (such as warehouse transfers) are being logged.

The system is automated up to a large extent. However, there is always a lurking potential for problems. Problems could occur while a box is being transferred since the overall operation is held within the range of external stakeholders. Successive internal changes could also affect a box: as unexpected box label removal, label destruction and even label misallocation. Some dangers could even be more dramatic such us: box destruction due to moisture, etc. Physical threats make more difficult the monitoring of boxes. Since the system deals to a great extent with physical artefacts there is exposure to dimensions which are outside the context of the system. This leads to uncertainty regarding what could be the exact status of an investigated box case. BTS system stakeholders would like to have elements of ambient reasoning while investigating a given box or a case (such as a loss complaint) from an existing vendor.

In an investigated situation they would rather prefer to be assisted in finding out what is the current context. Past experience could assist in such cases providing reasoning and potential scenarios for the investigated one. An intelligent monitoring system backed-up with a suitable CBR interface could definitely be of help in such case.

5.5.2 BTS and CBR-WIMS

The nature and the characteristics of the BTS system make CBR-WIMS suitable for its intelligent monitoring. BTS:

- is a workflow orchestrated system

- contains various human actors that deal with a large variety of tasks
- contains elements of uncertainty
- is heavily reliant to monitoring, etc.

As seen from above the monitoring needs of the BTS could be satisfied with the usage of CBR-WIMS. However, several changes have to be done in order to be able to accommodate the new system.

CBR-WIMS offers an agile architecture where several existing components and services can be (re) used in order to incorporate the BTS business process. Primarily the provided functionality of the kernel could enable the smooth integration with BTS, that it is why it was used accordingly. A new adaptor has also been created based on the business rules of BTS. This adaptor would enable the monitoring of the business process.

Since BTS is a new workflow system for the WIMS engine, past operational information has to be provided for the CBR component to operate. Additionally the range of conducted experiments with the BTS was not as broad compared to those conducted for the previous evaluation of the engine. The reason for that was that the research investigation was on whether CBR-WIMS can provide a generic intelligent system for workflow monitoring. The following sections will present the operational approach of CBR-WIMS towards BTS, the application of similarity measures among its transaction sequences and finally the produced results.

5.5.3 BTS Adaptor

Since CBR-WIMS has to be attached to a new workflow system, its unique characteristics have been considered. A box is an elementary unit of the newly investigated system, thus its explicit attributes had to be taken into account.

Each box within the BTS could be subject to a number of transfers also referred as journeys. These journeys have taken place throughout its lifecycle and could refer to:

- its initial transfer from a vendor to a designated warehouse
- a possible return back
- transfers among warehouses, etc.

Some boxes could have been transferred in bunches either because they were loaded to the same vehicle or because they belonged to the same vendor. All journeys consist of a number of stages, these can be referred to as events, and are being logged in a relational database. Events can be audited on demand by an authorised member of staff.

For the needs of the monitoring evaluation of the BTS, a new adaptor has been generated in CBR-WIMS. The adaptor was based on the BTS business process rules in order to be able to establish monitoring hooks with it. The rules have been extracted from the delegated component of CBR-WIMS, allowing further imminent communication with BTS. Relative controllers of the platform have also been adjusted to the business process based on the extracted rules and constraints.

BTS data were stored in a database format including some thousands of box transactions. The parse module of CBR-WIMS was used to extract the data and allocate them appropriately as cases within the system. Since this was a sensitive operation over a large amount of data, significant amount of time was spent on optimising the layout of the cases. Experience from the EMS investigated system has shown that such layout optimisation increases the proximity of past experience retrieval in its future uses.

5.5.4 Assessing the adaptability of CBR-WIMS

In order to demonstrate the adaptability of CBR-WIMS two sets of experiments have been conducted. The first one contained a limited amount of data, investigating the adaptability of WIMS to a workflow orchestrated environment. The second one included a significant amount of data, focusing on the efficiency of WIMS while monitoring a different workflow-orchestrated system. For the needs of the experiments a team of experts was consulted having experience in both the EMS and the BTS systems. The experts were used to rank the cases and their judgements were used to evaluate the CBR-WIMS performance.

Prior evaluations presented in this thesis (see the EMS section of experiments in Chapter 4) were focusing on whether the suggested research approach was feasible. However, the focus of these experiments is mainly to investigate whether the addressed research approach can be generalised among workflow systems with underlying uncertainty.

The experiments conducted with the BTS business process followed a similar methodology as presented in Chapter 3. Firstly the data from the investigated system were imported into the WIMS engine. The incorporated system contained more than seventy four thousand (74000) journeys that subsequently dealt with thirty three thousand (33000) unique recorded boxes. Each journey could be characterised by a range of events. The event attributes could vary from six (6) to more than a hundred (100).

Journeys at first stage have been incorporated and indexed inside CBR-WIMS as unique cases including several key attributes, secondary ones and relevant environmental information. The representation of events among cases was conducted via graphs in order to be able to apply similarity measures to them afterwards.

Time is of relevant importance while monitoring artefacts in BTS thus it was taken into consideration for the estimation of the similarity measures among its data. Maximum Common Sub-graph (as explained in Chapter 3) has been used, estimating the similarity among points and intervals. For this case study, events within the life of a box were regarded as points and the time periods among their transportations as intervals.

Initially the case base was divided into two parts in order to assess the adaptability of CBR-WIMS to the imported workflow system. For the needs of the artefact classification a number of interviews were conducted (BTS form can be seen in Appendix B) with the system experts in order to define the criteria for the classification mechanism of WIMS. The experts were asked to give a classification for the available workflow cases as well as support their judgement with free text comments, describing the faced issues of the case. Based on their given rankings the status of the boxes could promptly fit into one of the following categories: safely stored, damaged, lost and no clear status.

For the needs of the first experiment a case base of a thousand (1000) cases has been called to classify a randomly taken sample of 50 unranked cases. For the classification of the examined cases similarity measures were applied identifying the k-nearest neighbours of each case. The classification of any examined case has been conducted by a simple voting algorithm. For the needs of this experiment k was selected to be 3 due to the small volume of cases. In order to improve the proximity of results the experiment has been conducted 10 times and the produced results were finally averaged over the 10 runs. The results produced from the WIMS were afterwards compared with the classification of the human experts. Table 5-1 summarises the results of the experiments.

CASES	CBR-WIMS	Expert's classification
Correctly classified *no clear status* workflows	**13 – 68.4%**	**19**
Missed cases	**6 – 31.6%**	**0**
False positives	**4 – 13%**	**0**
Correctly classified normal workflows	**27 – 87%**	**31**

Table 5-1: WIMS simple evaluation with BTS.

The results gained from the conducted experiments have shown a relative maturity for CBR-WIMS towards the monitoring of a newly incorporated workflow system. A possible reason for that could be the way the system identifies and reasons over the imported cases. The more careful the selection of the cases is the more accurate the system is in its predictions.

Interestingly from the derived cases there has been no reference to cases with damaged or lost boxes. This was due to their seldom appearance in the overall data repository. Overall the results from the conducted experiment have been satisfactory having a positive classification of 68.4% for the unclear cases and a significantly high percentage of accuracy (87%) for the non-problematic workflows. Since this was a limited experiment in terms of the data used, a more advanced one was designed and launched taking into account the fully available data repository.

5.5.5 Assessing CBR-WIMS with full dataset

For the further investigation of the CBR-WIMS monitoring efficiency the full BTS dataset was used (more than 74000 journeys). This has populated the case base for the next experiment. In a similar approach to the first one a random sample of 500 cases has been selected as unclassified and the system was called to apply diagnosis-check on them. A rationale similar to the previous experiment has been followed, having applied prior similarity measures and having the k-nearest neighbours voting for the classification of a scrutinised case. Experiments have been conducted for 10 times (following the previous rationale) and the results have been averaged over the 10 runs. Table 5-2 summarises the results.

CASES	CBR-WIMS	Expert's classification
Correctly classified *no clear status* , *damaged*, *lost* workflows	**94 – 62.7%**	**150**
Missed cases	**56 – 37.3%**	**0**
False positives	**72 – 20.6%**	**0**
Correctly classified normal workflows	**278 – 79.4%**	**350**

Table 5-2: WIMS evaluation with the overall BTS data repository.

As it can be seen from table 5-2, WIMS has been shown efficient with the classification of the BTS dataset. Normal cases were able to be identified in 79.4% of the times and the system was accurate in 62.7% of all the other cases. These results have been regarded as satisfactory from the BTS experts. Due to the volume of the cases and the provided information a need was formulated for amplified reasoning, especially for the explanation of the WIMS classification. As a result an Agglomerative Hierarchical Clustering (AHC) algorithm has been used, similar to the one discussed in section 4.5, to enhance the confidence of the users regarding the provided results.

Generally workflow users and stakeholders need convincing arguments in order to increase their trust to an insight provided by an engine. Therefore AHC has been used extensively in the BTS experiments to amplify the provided reasoning and explanation by being able to drill down to cases and show their similarities along with any related knowledge. The adopted approach regarding effective reasoning will be discussed at the following chapter.

5.6 Conclusions

This chapter investigated how the proposed research approach can be applied across workflow-orchestrated and choreographed business processes. It is profound that in order to be able to provide intelligent monitoring to a complex workflow-orchestrated environment, agile software engineering has to be applied. The main reason for that is that it should be able to provide support for constant changes, adjustments within a business process and constantly developing business needs.

CBR-WIMS has been designed as a generic workflow monitoring framework which facilitates a number of flexible services, control modules and adaptors. CBR-WIMS can offer an interoperable environment for workflow monitoring and diagnosis. This chapter has presented the architectural backbone and has revealed how its collaborative components could accommodate that. An alternative system orchestrated, choreographed and operated via a workflow has been used as a test bed for the evaluation of CBR-WIMS. WIMS have shown efficient in terms of its adaptation capabilities on the different system.

The applicability of CBR-WIMS to different business processes has shown that it can establish intelligent workflow monitoring. An interesting finding from its application on BTS is that heedful knowledge acquisition leads to its reusability across workflows.

CBR-WIMS has revealed how it can be expanded, adapting its monitoring functionality across systems. However, the provided functionality was accompanied with a number of explanation elements that enhanced its monitoring, reasoning and diagnosis on workflow systems. These will be presented in Chapter 6 along with how explanation is tightly related to knowledge elicitation across systems.

Chapter 6

Explanation

6.1 Introduction

Explanation is an important field to intelligent systems. An approximate definition of explanation in systems is the collection of logical arguments that could convince a user to adopt proposed assumptions, recommendations and even decisions up to a certain extent. In artificial intelligence, and especially in the area of case-based reasoning, systems are able to extract explanation knowledge from the available past information. This can be presented afterwards to their human stakeholders in order to provide reasoning and justification for any system recommendations and/or decisions.

The explanation provision increases the confidence in an artificial intelligent system and has as a target the development of trust between itself and its users. Everybody's personal experience shows that building trust among humans is a difficult task. Therefore if this task is applied to an artificial environment it can be even more difficult.

This thesis investigates whether there can be intelligent monitoring of business processes using case-based reasoning. A process of artificial monitoring is tightly connected with the provision of profound explanation. In any different case a system that does not explain its outputs it is difficult to be tangible and understandable by human stakeholders.

As it has been shown in Chapters 1, 3 and 5 the process of workflow monitoring requires explanation provision in order to enhance its reasoning, diagnosis and possible remedial suggestions / recommendations. Therefore the provision of effective explanation has been investigated throughout this thesis. The focus was on what constitutes an effective explanation while monitoring a workflow process and how it should be (re)presented to the users of the system.

This chapter refers to explanation as derived within the context of the conducted research. Its structure is as follows: Section 6.2 gives an overview of the explanation concept; Section 6.3 gives an outline of explanation in applied systems; Section 6.4 discusses the concept of explanation in workflows and Section 6.5 refers to the concept of explanation in CBR-WIMS platform. Finally section 6.6 presents the conclusions of this chapter.

6.2 The concept of explanation

Explanation presents a prominent topic in science [Schurz, 1995], often referred to as scientific explanations. Scientific explanations attempt to give answers to why questions by using existing facts and applying general laws. However, these answers can be different regarding the application domain and as a result explanations can differ. Also there is a distinction between cause giving explanations and reason giving justifications [Schurz, 1995], referring to the equivalent ‘why’ questions respectively. The first category of the above, questions why something has taken place like that, what was its cause or reason for being. An example of that could be:

Child’s Question: Why is the water in the lake frozen?

Father’s Answer: Because the water temperature is below 0^{o} Celsius.

The second category examines what was the reason behind the occurrence of an event. Explanations of this category could be referred to as identifying what is the reason for believing why an event has happened in a specific way. An example follows:

Wife's Question: Why *didn't* you call me the whole day?

Husband's Answer: Because I had too much work in the office, darling. I *didn't have the time even to get some lunch...*

Although explanation seems a self-descriptive concept which reveals the answer to intended knowledge questions, in reality explanation can be literally false [Roth-Berghofer, 2004]. This is being done deliberately due to moral, pedagogical, social and other context-dependent reasons [Cohnitz, 2000]. These explanations are being provided in the context of satisfying the questioner and not necessarily fulfil the purpose the questioner expects them to. An example can be a question from a seven-year old girl to her mother on how her little brother came to life. The most prominent answer that she could get is that the stork brought him home, when everybody was asleep or some variants of the above. Explanations are heavily dependent to the background context of a given environment.

Schank (1986) has characterised explanation as the most common method used by humans to support understanding and decision making. An explanation should describe a solution to a problem as well as which is the path that has to be followed in order to reach the solution. Therefore explanations are characterised as both inclusive and instructive [Roth-Berghofer, 2004]. A system can explain its actions both to humans and/or services that inquire how it works as well as to itself. In such way, according to Schank, it becomes an understanding system. The range of cognitive understanding of such a system can vary from making sense to complete empathy [Schank, 1986]. Computer reasoning stands towards the making sense edge of the above range trying to convince its users rather than blindly guide them.

Explanation, according to Schank, can be distinguished to three main classes. These are the physical world, the social world and the individual behaviour patterns. An artificial agent can give sufficient reasoning to a case that fits in to one of the above classes. Especially for the first class of explanation (physical world) reasoning can be derived from the laws of physics. Having those as a basis, more complicated explanations can derive for other (presumably) more complicated fields. Explanations for the social world and behaviour patterns seem the most complicated ones being of relevant importance in the modern world. For the needs of the explanation to those areas advanced techniques like data mining, user profiling and machine learning could be used. These techniques attempt to extract the followed behaviour patterns in an investigated discipline.

The reasoning provision by itself is not enough to convince a human user for the clarity and decision accuracy of a system. Therefore for a monitoring system to be efficient its individual users should develop trust in it. Intelligent systems should provide meaningful explanation while presenting results or suggestions in order to increase the user confidence to them. To achieve that evidence for any specific system, output should be presented to its users in order to provide appropriate justification for the output. Users are always convinced more for the accuracy of a system when next to an output (no matter if it is good or bad), evidence is presented on how this output was derived [Swartout, 1983].

While attempting to formulate explanation certain goals should be fulfilled. To achieve that, Sørmo et al. (2005) have identified five distinct explanation goals. These goals either isolated or in combination could be used to construct explanations. These are:

- How did the System reach the answer? (Transparency)
- Explain why the answer is a good answer. (Justification)
- Explain why a question asked is relevant. (Relevance)
- Clarify the meaning of concepts. (Conceptualisation)

- Teach the user about the domain. (Learning)

Usually explanation systems are satisfying one or more of the above goals while attempting to provide sufficient explanation.

6.3 Explanation in Systems

As seen from the previous section explanation is needed to answer questions regarding the cause, reason of existence or result of an occurring event. Nowadays artificial systems are being used in the area of problem solving, in a similar way to human experts. Confidence has to be built with the human users in order to be at a position to decide whether to accept or reject an artificial recommendation. The user has to be able to drill down to a specific system-output and be able to scrutinise, if necessary, what led the system towards that direction. Due to the need for sufficient reasoning, a need for providing efficient explanation is being increased respectively.

The explanation that can be extracted from a system is based on the nature and the contextual background of the system itself. An example can be an Artificial Neural Network which by its nature contains the knowledge inside its internal structure. As a result it works as a black box and cannot provide sufficient explanation to support its outcomes since its knowledge cannot be extracted. A system that uses genetic algorithms to calculate its output and propose recommendations faces similar explanation limitations. Rule-based systems can perform better since they can resort to the reasoning provided by their rules. However, several restrictions apply: In order for this to be efficient, the domain should be limited for the user to be able to follow and evaluate the explanation. In any different case the complexity of the correlated rules can be unmanageable since even experienced users are not usually able to follow such stated explanations [Roth-Berghofer, 2004]. Another limitation rule-based systems face is the significant growth in the number of rules that they have to contain in order to operate in a complex domain. This number, as well as the effort for their acquisition, makes them a relative complex domain for providing satisfactory explanation.

Case-based reasoning systems could offer a solution to the faced limitations of ANNs and rule-based systems. This is due to their ability to present the information of their cases in order to provide support to a certain system output. Therefore cases can be descriptive in terms of a problem and its surrounding information. However, cases do not provide information in terms of their selection criteria and the rationale behind that. As a result a different entity is required that connects the related knowledge to a case and its available structure. This is can be related to the concept of knowledge containers.

Knowledge containers, introduced by Richter (1995), refer to both the knowledge in cases as well as the structure of that knowledge. Due to its structural generalisation, a knowledge container can refer a number of tasks combining their related characteristics to possibly one schema. Richter has introduced four (4) knowledge categories for CBR systems. These are presented in Figure 6-1 and are: the vocabulary, the similarity measures, the solution transformation (or adaptation rules) and the case base.

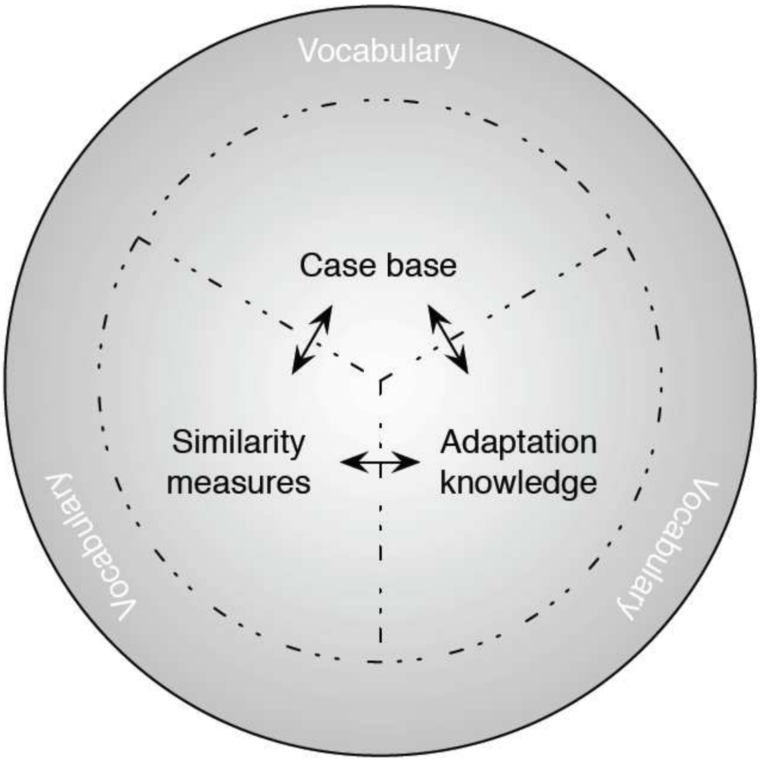

Figure 6-1: The four knowledge containers of a CBR system [Roth-Berghofer, 2004].

The Vocabulary container is the basis for the other three since it contains everything that defines a system: attributes, predicates and even the domain structure. The Similarity measures container contains critical knowledge on how the similarity is calculated. This is important since it can give answers to the questions why and how the most useful case is retrieved. The Adaptation knowledge accommodates the knowledge of how an old case is adapted to meet the requirements of a new case. Finally, the case base encapsulates the available past knowledge of the system in terms of cases.

Knowledge in CBR systems is being deployed in a structure that enables smooth acquisition. Since the cases are specific, their relations can be easily identified, generalised on demand and then modified accordingly in order to be reused. The operational domain of a case is also limited in contrast with rule-based systems where the knowledge domain is general. By organising the CBR knowledge into containers the system can be more flexible and reusable since changes on one container have little impact to others. For example the addition of new cases does not affect the similarity measures and this “helps maintaining the knowledge of case-based reasoners” [Roth-Berghofer, 2003]. Changes in one container always have some or little impact to the others but their distributed approach leads to better knowledge management. Finally knowledge can be shifted among containers either artificially or manually based on the context of the CBR system.

Explanation in CBR can be compactly associated with the tracing of provenance in systems. By default if the provenance is known for a specific event it leads to constitution of better explanations. Mantaras et al. (2005) have identified two different categories in provenance: The first category is regarding the provenance capture as a possible source for the mining of cases, whereas the second one tracks provenance within the CBR systems in order to use their cases in a better way. The above provenance categories can be regarded as external and internal respectively.

CBR systems seldom remember how their cases have been derived [Leake, 2008] within their execution time. Recent research work has started focusing towards the usage of internal provenance information for several tasks. Such a task could be the recovery from delayed feedback by propagating feedback information to related cases. This can predict the case quality based on quality loss through repeated adaptation and using provenance to target maintenance [Leake & Whitehead, 2007; Leake, 2008]. This available internal provenance can be used to guide its related CBR process. By using provenance the maintenance is more powerful than the ordinary maintenance, which detects and fills gaps or responds to problems caused by inconsistencies. The provenance approach presents an active role by suggesting a priori possible candidates for case replacements or confirmations [Leake, 2008].

Explanation in CBR systems is tightly related with the built of trust and confidence among their users. Monitoring systems that use CBR for their reasoning and explanation are tightly connected with the explanation provision in terms of context, relevance and the provenance of their cases. In this thesis explanation has a vital role since it deals with the monitoring of workflows with present uncertainty. Provenance was also used to build trust and to provide reasoning based on the fundamental assumption that similar problems have similar solutions. The following section will get in more detail regarding the explanation in workflows and the conducted experiments for this research will follow.

6.4 Explanation in workflows

Business processes are being represented successfully in a number of standards like BPMN and BPEL as shown in Chapter 2. By using standards like WfMC and XPDL business processes can be standardised and exchanged among systems and workflow products. Business process designers, developers, managers and generally actors can use the above standards to define and understand the context of business workflows. This can be done with relative ease since the design of the existing standards allows it. However a problem occurs while trying to monitor an executed workflow. The reason for that is that a human expert should resort to the information produced from the workflow in order to be able to establish a possible monitoring procedure.

In order to be able to monitor workflows, the human experts have to look through the information produced from an executed workflow. This information exists in the form of sequential events that took place during the execution of the workflow. These events are usually saved in operational logs within the system. Logs can vary in size based on the information captured as well as the content of the information. This has heavily to do with the context of the operational workflow: a patient diagnosis workflow is expected to have different content from a sales management one. Usually in well-audited or large systems the found logs can be extremely large. On top of that the more their facilitated system increases in size and operational density the more their actual complexity increases. This affects directly their ease of readiness, makes more difficult their understanding, validation and finally their monitoring.

Most workflow logs nowadays are well structured and small pieces of structured texts can be easily understood [Michaelis et al, 2009]. However, if their size rises significantly the mental capabilities of a human are not enough to deal efficiently with them. In addition to that a specialised workflow structure may deprive the expert from interpreting effectively the current workflow state. This could be due to the necessity for more specialised auditors for specific parts of the workflow log.

Apart from the possible problems with log size and complexity in data, the issue of temporal relationships can make difficult the job of an auditor. Correlated events with many or overlapped temporal relationships are difficult to be monitored. Furthermore for effective monitoring a combination of a number of logs may be needed and/or their possible association with events that may not be captured [Kapetanakis et al., 2010b].

Workflows are often adapted or manually overridden in order to deal with unanticipated problems and changes in their operational environment. These changes are more frequent in the aspects of workflows that interact primarily with human roles. This can lead to incomplete contextual knowledge and possible uncertainty. Uncertainty is very possible to occur in workflows since business requirements change often. They may also involve processes from different parts of an organisation or parts from collaborative organisations which may cause conflicts.

As can be seen from the above, the monitoring of workflows and their possible diagnosis can be a difficult and complicated process for human experts. Therefore the need for artificial interpretation and monitoring of a workflow are being formulated. An artificial system that attempts automatic workflow monitoring should provide the ability to provide explanations to the human auditors of the process. These explanations would allow them to identify, understand and act on possible problems identified in the execution of the workflow. In order for this to be efficient the system should be able to provide adequate arguments on why a particular workflow trace has been highlighted for attention.

Explanation should be additionally provided while reasoning on why a workflow requires intervention, what could be the identified causes as well as what remedial action may be required. A step towards that direction could be the provision of proximity or similarity in terms of the visualisation of a case. In this way the user of such a system could see the problem and possibly be persuaded for the indicated further actions. If s/he is not convinced then s/he could proceed with a possible override or adaptation of the workflow.

Workflows change over time, therefore looking at the provenance of a workflow instance can provide a better understanding when extracting insights. This in correspondence can make cases more / less usable. Provenance can be significantly important when trying to reason across business workflows. A reason for that is that provenance can provide reasoning to the whole spectrum of the steps followed towards a known outcome. This could lead to reusability of the acquired knowledge across a number of similar cases.

An example demonstrating the importance of provenance and its mighty reasoning reusability could be taken from the area of financial / banking transactions. It is common for financial organisations to offer loans to their customers. By having a loan system working for a period of time, a certain amount of knowledge is being acquired from the process. This knowledge can contain several cases including loan approvals and rejections along with complementary knowledge cases (for example incomplete cases). If provenance exists in these cases it allows knowledge reusability across a similar loan procedure in a different / subsidiary organisation or even to a different process, e.g. a mortgage approval one.

6.5 CBR – WIMS & Explanation

This thesis investigates whether there can be efficient artificial monitoring and diagnosis on workflows using CBR. A certain characteristic of this approach is the fact that a workflow monitoring system that uses CBR does not attempt to build an explicit model of the knowledge associated with workflows. What it does is to provide reasoning on the fundamental assumption that similar problems have similar solutions [Kapetanakis et al., 2010b]. Since a similarity measure has been defined, it can be used afterwards to identify possible solutions for a specific target workflow case. Similar neighbour workflow cases are being retrieved from a case base and their encapsulated solutions could be used to provide a solution to the investigated workflow.

The explanation required for such a system is the one that can show to its users why the retrieved workflow cases are regarded as mostly similar. However, this is something that can differ based on the different perspectives of the same problem. Atzmueller & Roth-Berghofer (2010) have identified six (6) areas of explaining that refer to different perspectives of a problem. These can deal with the:

1. "Different explanation goals
2. Different kinds of explanation
3. Modes of presentation
4. Level of detail of explanation: concrete vs. abstract
5. Utilisation of different knowledge container.
6. Privacy: Which data/information or knowledge from the different knowledge containers is actually revealed to the user?" [Atzmueller & Roth-Berghofer, 2010]

Explanation should be able to present the selection criteria as well as show how the retrieved cases can be used to provide a solution to the investigated problem. In a number of typical CBR systems, similarity and relevance of neighbour-cases to the investigated one can be simple to understand. However, providing a suitable and trustworthy explanation to workflows is a challenge due to their structural and temporal complexity.

Within the adopted research approach explanation is needed to support the intelligent monitoring of business workflows since it is rather important for a business process manager to be able to clearly understand the actors and their actions. As a result visualisation techniques were applied when conducting the experiments, containing real workflow data and using workflow similarity measures (See Chapter 3). Aiming to an enhanced reasoning provision the architecture of this approach (as described in terms of CBR-WIMS and presented in Chapter 5) was designed with relevant emphasis to adequate explanation provision.

CBR-WIMS was built with the aim to provide automatic monitoring to workflow stakeholders. It can identify potential problems and provide advice on actions that can remedy an observed problem. The system is based on the experience from past events, their contextual knowledge and their available classification. The applied similarity measures allow the retrieval of close matches as well as their associated workflow knowledge. This subsequently allows the classification of a sequence as a particular type of problem that needs to be reported to the monitoring system. Additionally any associated knowledge or plan of action can be retrieved, adapted and reused in terms of a possible recommendation / suggestion for a remedial action on the workflow.

6.5.1 Visualisation in CBR - WIMS

As it has been illustrated so far the architecture of CBR-WIMS has been designed with significant emphasis put on the explanation provision. In order to be able to provide sufficient explanation, the proposed suggestions have to be presented in a way that promotes trust.

Therefore their effective visualisation has been broadly considered and presented in an obtainable way.

This section will present the routes in which the engine promotes explanation via highlighting the relevant areas that had to be considered. The experiments that were conducted in order to evaluate this explanation provision will consequently follow.

Since CBR-WIMS deals with the monitoring of business processes, newly investigated cases have to be compared with the ones available at the existing repository of cases. An expert that would like to drill down to a particular system output should also be able to drill down to its individual similarity measures. The way the result has been calculated as well as the relevant information that was taken into account has to be presented in order to increase the reliability of the system.

When comparing two workflows, the similarity is measured as the normalised sum of similarities among the non-overlapping MCS segments in the two workflows. The identification of these workflow segments and their representation to the user allows the visual explanation of the CBR monitoring process to:

- Be explicit enough, explaining visually how similar two workflows are
- Indicate visually which parts of the investigated workflows contribute most to this similarity. This allows the user to focus to specific patterns in parts of the workflow that may flag it as problematic, unstable or on the contrary stable and faultless
- Visualise past remedial actions that were applied as well as the results of such actions. This can assist a user to understand why the system has made a particular diagnosis and provide insights into potential remedial options / actions to the target workflow case

Figure 6-2 below shows the visualisation of the similarity between simple linear workflow event logs. Similar workflow segments as estimated by the MCS are being highlighted. These segments provide an ability to drill down to the individual workflow segments, events and intervals. The similarity measures and diagnosis as well as the proposed actions can also be shown on demand. As a result this can provide to a user a deeper understanding of the workflow monitoring process.

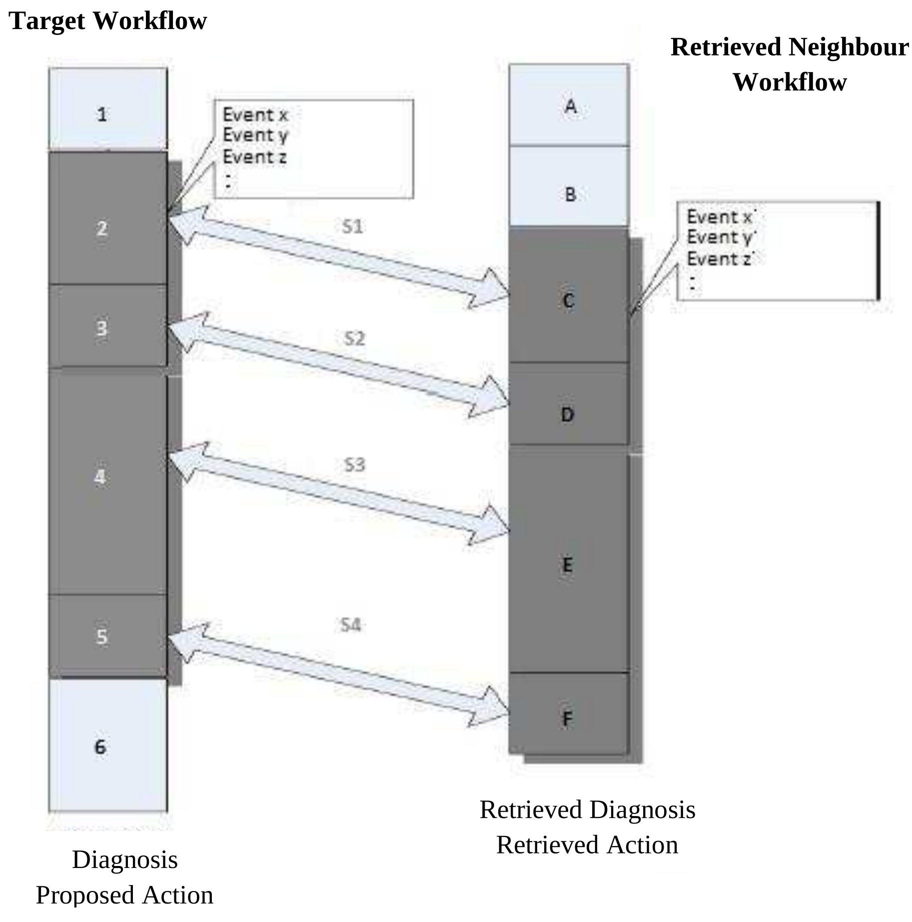

Figure 6-2: Visualising the similarity between workflows [Kapetanakis et al. 2010b].

Effective visualisation of the conducted similarity measures among workflows can provide an in-depth reasoning regarding the foundation that led the system towards a certain decision. While the volume of the case base rises, a better reasoning provision is required in order to provide its users with a comprehensive overall (more abstract) picture. Therefore the system offers a hierarchical clustering visualisation allowing its users to understand how an investigated case fits into clusters (formulated by existing past experience).

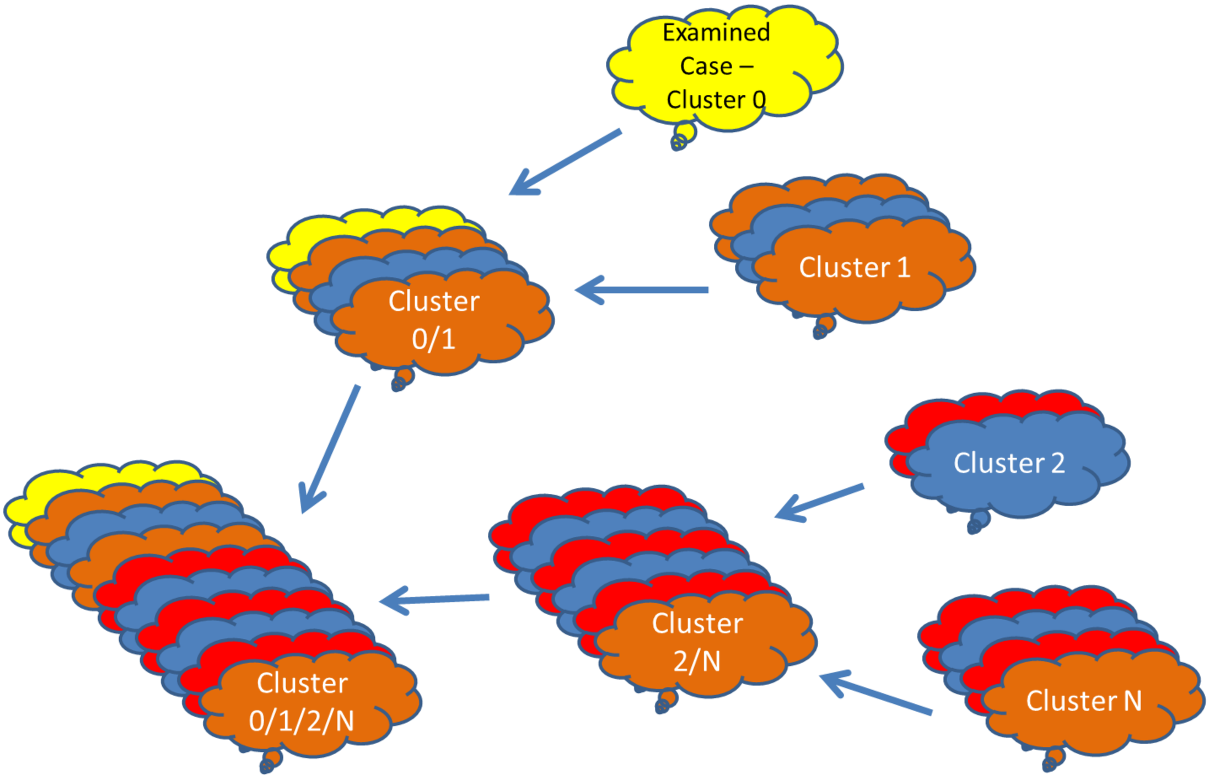

Figure 6-3: Application of AHC to an examined case.

Figure 6-3 highlights how an examined case could fit into existing clusters containing past experience. The division of cases into clusters gives a significant advantage to experienced system stakeholders to identify cases with similar characteristics. This was triggered from the assumption that when a manager of a business process has identified a problematic behaviour in an examined case, a certain type of pattern is identified. This is based on the acknowledgement that certain types of patterns can identify certain types of problems. By applying clustering analysis, there is a tendency to group together the cases with similar patterns, therefore identifying the cluster(s) with potential problematic behaviour.

CBR-WIMS deals with systems that contain temporal complexity. In order to be able to reason upon that a hierarchical cluster analysis algorithm could be applied as seen in section 4.5, rather than possessing a static set of values for each case. The application of the Agglomerative Hierarchical Clustering (AHC) algorithm has been regarded as the most appropriate since it can organise its clusters in a progressive way based on their similarity distance.

In such way cases are represented as clusters and those with the highest similarity between them are sequentially merged into a single one. The cluster merge is being conducted in a progressive way allowing the system experts to have an enlightened overview of the cases as a whole. The reverse process is also allowed: to split an already framed cluster into its elements and rationalise upon the way it was formulated. Such approach allows a deterministic argumentation on how a cluster has derived revealing in parallel elements of its provenance. In such way the confidence of the experts to the suggestions of the system can be reinforced, raising the levels of trust to the system.

For the needs of the evaluation of the WIMS explanation capabilities some explicit experiments were conducted. The following section refers to them in detail.

6.5.2 Explanation provision in CBR-WIMS

The focus of the experiments was on the explanation provision in a wide range of facets including the similarity measures, the retrieval of cases, the clustering of cases and the final recommendation provision.

As a first approach towards the evaluation of CBR-WIMS explanation capabilities, small datasets were used to evaluate the efficiency of its visualisation components. Motivated by that a sample of 20 exam moderation workflows were selected, monitored and afterwards compared to a case base of 130 known-state workflows. The known-state workflows contained expert monitoring classification as well as remedial information.

The ability of the system to provide suitable monitoring advice on this dataset has been evaluated and already discussed in Chapter 4. The purpose of this experiment was to evaluate the ability of the system to explain the decisions of the CBR workflow monitoring system to workflow stakeholders. For the EMS system these were senior members of staff that were in charge of the process and for the BTS the managers of the archiving process.

Workflow stakeholders used WIMS-CBR to establish whether the system provides useful explanation upon the areas of:

a) Correct classification of a workflow (stalled / not stalled)
b) Reasoning and diagnosis for the stalled cases
c) Proposed remedial actions

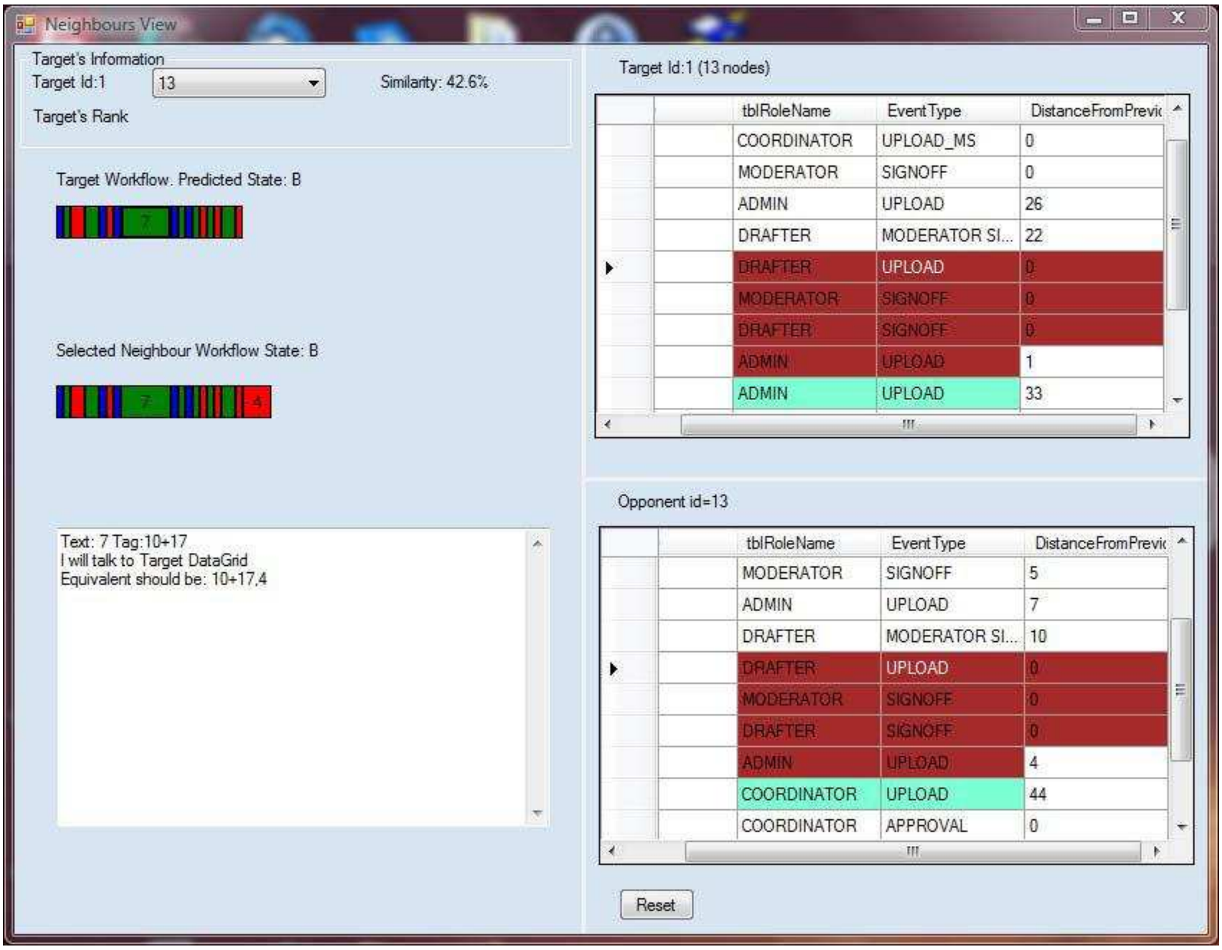

Figure 6-4: The Similarity explanation in CBR-WIMS [Kapetanakis et al. 2010b].

Figure 6-4 shows a typical view of the explanation screen of CBR-WIMS. The user can select any of the nearest neighbours of the target case and investigate its inner structure, the similarity among cases as well as any retrieved advice. A colour visualisation has been selected for highlighting the regions of similarity as well as the degree of similarity. The visualisation was defined as follows:

- Green indicates 100% similarity
- Blue indicates similarity from 0.001% to 99.999%
- Red colour indicates 0% similarity

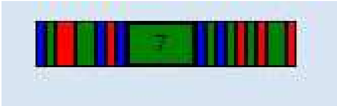

Figure 6-5: Colour code, indicating levels of similarity.

The system allows users to click on any particular matching workflow segment within their MCS structure. This permits them to visualise any overlapping areas which are automatically dyed in the same colour and presented on the screen. Users are allowed to drill-down to particular areas of similarity among workflows and investigate the monitoring advice retrieved from their closest neighbours.

For the need of this experiment the team of experts from the EMS and BTS systems as seen in section 5.5.4 have contributed with their experience to the evaluation of the CBR-WIMS explanation provision.

Table 6-1 summarises the results of the simple evaluation of the explanation capabilities of CBR-WIMS. The experts were called to assess the monitoring findings of the system with and with-out explanation provision. In order to measure their confidence a scale of 1 (disagree) to 5 (strongly agree) was used indicating how convinced they were regarding the presented findings. The results were averaged over the 20 target cases.

	WIMS-CBR no explanation	WIMS-CBR with explanation
Correct classification is clear	3.2	4.2
Similarity is obvious to the 3NN	2.8	3.9
Advice clarity	3.3	4.5

Table 6-1: Evaluation of the explanation and advice [Kapetanakis et al., 2010b].

This evaluation has shown that the explanation module in CBR-WIMS can provide a significantly better insight to workflow stakeholders. From the conducted experiment it has been testified that the explanation provided allows the system experts to understand better the monitoring warnings as well as to get a clear insight of the process. This raises the confidence levels on the associated advice regarding any remedial actions. It is worth noting that even in cases that the CBR system retrieved low quality solutions (mainly false positives), the experts reported positively towards the system. That was due to the explanation provided since it allowed them to discard the retrieved solutions. The drilling down to specific workflow event data showed that the retrieved advice was not relevant to the problem in hand. As a result it affected directly their following decisions.

Motivated by the same attitude (evaluation of the explanation provision of CBR-WIMS) a set of similar experiments was expanded and conducted on the BTS workflow system (discussed in section 5.5.2). The allocation of a different workflow system under the monitoring hooks of CBR-WIMS gave a different angle to its explanation provision. Explanation facilities based on the generic Explanation component of CBR-WIMS were adjusted and adapted to the new system. These alterations have been conducted in order to be able to provide any extracted insights from similarity measurements on different cases.

In order to be able to provide sufficient reasoning the focus of the explanation was concentrated firstly on the similarity measures and secondly on the overall visualisation of the cases. For this set of experiments a sample of 30 box-workflows was selected, monitored and afterwards compared to a case base of 180 known-state workflows. The known-state workflows similarly to the previous example contained expert monitoring classification as well as any remedial activities.

Figure 6-6 shows the visualisation and explanation provision for a sample of event sequences in BTS workflow system. As it can be seen a number of attributes have been taken into consideration and several individual measures constitute the final similarity result.

166450	Normal	05/04/2007 18:18	Committed	539	PWD	Manual	Delivered	Initial Transfer, Permanent Withdrawal	TRUE	FALSE	30/11/2005 14:16
166450	Normal	?	Committed	539	PWD	Manual	Delivered	Initial Transfer, Permanent Withdrawal	TRUE	FALSE	30/11/2005 14:16
166450	Normal	15/11/2007 14:12	Committed	15364	PWD	Manual	Delivered	PWD by stored proc		FALSE	16/05/2008 00:00
166450	Normal	15/11/2007 14:12	Committed	15364	PWD	Manual	Delivered	PWD by stored proc		FALSE	16/05/2008 00:00
166451	Normal	05/04/2007 18:18	Committed	539	PWD	Manual	Delivered	Initial Transfer, Permanent Withdrawal	TRUE	FALSE	30/11/2005 14:16
166451	Normal	15/11/2007 14:12	Committed	15364	PWD	Manual	Delivered	PWD by stored proc		FALSE	16/05/2008 00:00
166451	Normal	05/04/2007 18:18	Committed	539	PWD	Manual	Delivered	Initial Transfer, Permanent Withdrawal	TRUE	FALSE	30/11/2005 14:16
166451	Normal	15/11/2007 14:12	Committed	15364	PWD	Manual	Delivered	PWD by stored proc		FALSE	16/05/2008 00:00

Figure 6-6: BTS visualisation provision

Experts that were called to comment on the explanation provision came back with positive responses regarding the trustworthiness of the system on the provided results. Reasoning upon the results with the application of relevant visualised explanation has enhanced their confidence and led to faster classification of the investigated cases. Table 6-2 summarises the results from the actual visualisation of the conveyed explanation. The experts based on the system provisioning gave response to questions using a scale of 1 (disagree) to 5 (strongly agree). The results were averaged over the 30 target cases.

	WIMS-CBR no explanation	WIMS-CBR with explanation
Correct classification is clear	3.0	4.1
Similarity is obvious to the 3NN	2.5	4.0
Advice clarity	3.2	4.2

Table 6-2: Evaluation of the explanation provision on BTS

Based on the experiment findings it could be concluded that the visualisation provision enhances the confidence of the experts to the monitoring recommendations. However, this can be said with regards to isolated cases in comparison. While examining a case overall it would be better to provide a rationalised overview based on the classification of the case among others. Therefore a generalised approach via clustering has been provided and is being discussed in the following section.

6.5.3 Clustering

Drilling-down to individual similarity measures among stories enhances the explanation and builds confidence with users while using the system. However, it does not provide insights on the nature of the retrieved cases. Initially the implemented system was expecting from its users to be able to extract the underlying context from the raw, retrieved cases. However, observations on the behaviour of experienced managers lead to the identification of certain types of patterns which could indicate problematic situations.

In order to be able to identify such problems based on their primarily followed event pattern cluster analysis has been applied. A Hierarchical Clustering (HC) algorithm has been chosen in order to be able to provide comprehensive reasoning regarding the individual characteristics of the cases and their cohesion as a set overall. In order to deal with the sparse nature of the case-base, an Agglomerative Hierarchical Clustering algorithm was chosen, exploiting its ability to organise the clusters hierarchically based on their similarity distance. The reasons for this have been described in detail in section 4.5 therefore this section concentrates on its explanation aspects.

The main focus of the explanation provision via the AHC was to be able to identify any event pattern followed by the nodes of a selected cluster. The explanation component of WIMS promotes this behaviour by:

- Providing the most prominent pattern followed by information regarding related cluster nodes.
- Provide patterns identified in individual nodes
- Provide frequent sub-patterns among them.

Figure 6-7 shows how WIMS can represent clusters along with their patterns, sub-patterns and relevant information.

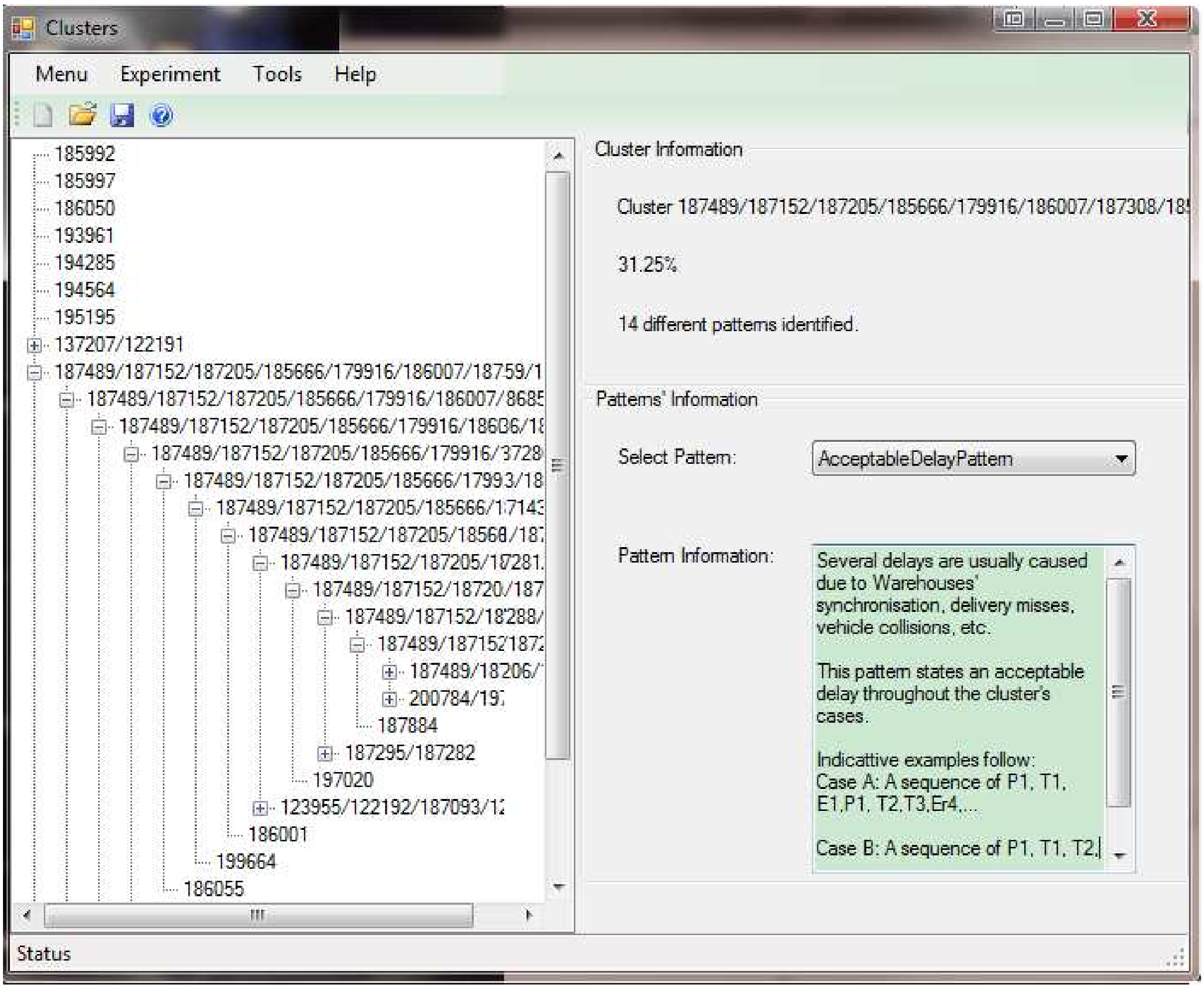

Figure 6-7: Cluster selection along with included patterns.

Since the clustering is being conducted via a hierarchical clustering algorithm, the whole case base along with the investigated case can be visualised and the breakdown of the followed steps can be seen. This breakdown could possibly reveal hidden patterns among clusters that are of importance. An expert could find this of benefit since s/he can understand better what is the relation of the cases, which are the similarities, which are the patterns that indicate problem or on the contrary a smooth operation, etc.

The usage of clusters in WIMS allows the user to compare an investigated case with available past cases taken from the case-base. Similarity measures are also provided allowing the user to understand why and how the system was driven to a particular diagnosis. By presenting all relevant parts of the matched cases a user can then see possible identified issues. Provision of the related contextual information can also lead to solid justification for a decision been taken.

The clustering provided by the WIMS enables workflow managers to drive their actions accordingly based on the evidence presented to them. The ability of the system to identify specific event-patterns, potential problems and the proposition of possible solutions enhance further its explanation capabilities.

Clustering experiments have also been conducted on the BTS workflow orchestrated system presented in sections 5.5.1 – 5.5.5. A characteristic of the investigated system was the significantly high number of attributes that should be taken into account for its similarity measures. The produced clusters were affected by this characteristic pointing out some unanticipated results. The system experts when called to comment on the monitoring outcomes heavily were based on the WIMS explanation facilities to point out why the system had these unexpected outputs.

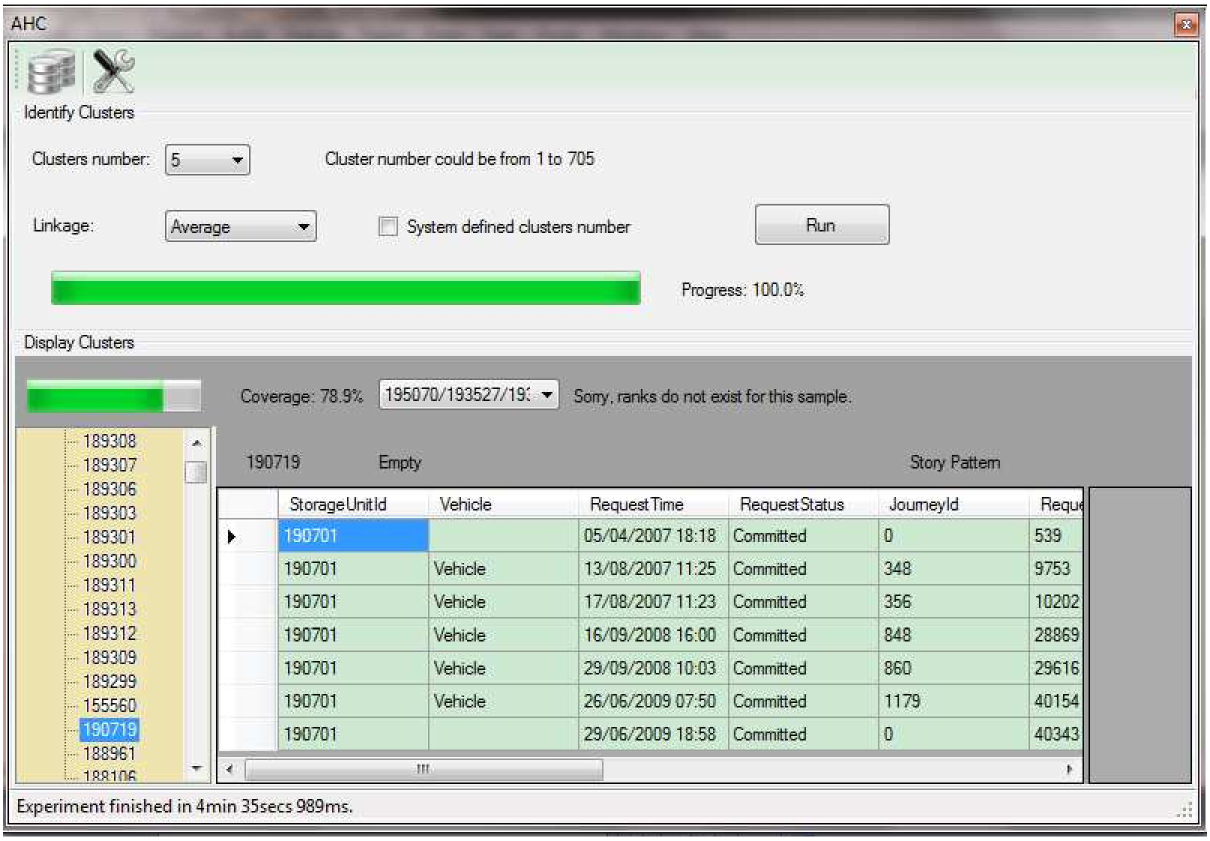

Figure 6-8: AHC applied on BTS

Figure 6-8 presents an AHC clustering visualisation as provided from CBR-WIMS. Clustering provision in explanation enhances significantly the confidence of the experts since it gives the opportunity to see the broader picture and afterwards drill-down on demand. Clustering, as it can be seen from above, gives the ability to generalise on the followed patterns of events and be used afterwards if there is a demand for further reasoning in similar cases.

6.5.4 Challenges in explanation provision

The WIMS explanation component is characterised by certain adaptability regarding its provided capabilities and its approach towards the investigated systems. However, throughout its application several challenges have been encountered. This was mainly triggered due to the nature of the investigated systems.

A first encountered challenge was the successful adaptation of the original component as provided by the framework. Each investigated workflow system poses a unique case for an intelligent monitoring system and its characteristics should be earnestly taken into account. The successful adaptation of the Explanation component in order to be able to extract the relevant contextual information has stated a sole challenge while designing and underpinning the system.

Another challenge of different nature was encountered while attempting to provide explanation regarding the similarity measurements among cases. Especially in the case of the BTS system the large number of available characteristics constituted a difficult region to explain to a human expert. This number of fields was creating a vague multi-dimensional set which had to be provided in terms of the output explanation. In order to meet this challenge, careful application of weighing among the available fields has been shown useful since it managed to emphasise on important fields and conceal the minor ones.

The final challenge encountered while providing explanation was the elicitation and actual presentation of the shadowed patterns followed by the investigated sequences. The reason for that was that certain patterns contained several redundant events which were adding noise. This noise could actually mislead the engine while attempting to reason upon a requested case. The sanitisation of events (see section 3.6.3) by smoothening their sequences and bound them to their most important features had helped significantly in overcoming this challenge. In such cases the users were notified and their permission was sought in order to proceed with. In any case that the users were interested in investigating the whole range of events, including any noise that was omitted, they could do so since the actual event sequences are kept in the system for maintenance reasons.

6.6 Conclusions

Intelligent monitoring systems deal extensively with the two parts of a business process: human actors from the one side and digital data from the other. Since their goal is to provide recommendations, proposals and/or solutions to their users, explanation is needed. The need of explanation is important in Decision Support Systems (DSSs) since they need to support with objective arguments their suggested outputs. This can be done by retrieving and presenting similar incidents from the past experience of a system when compared to an investigated case.

This chapter has presented how explanation resides within CBR-WIMS, how it is formulated and presented in terms of visualisation components. CBR-WIMS, since it is an intelligent workflow monitoring system, deals with complicated temporal past evidence that has to be categorised to cases and be accurately classified. The adopted approach in this thesis gives the system the opportunity to resort to the provenance of the cases. This approach enables the accurate handling of cases under investigation, since provenance provision can enlighten significantly the monitoring process. As a result this is reflected to the recommendations / suggestions of the system enhancing its monitoring capabilities overall.

CBR-WIMS is able to explain the context or the probable cause of a problem through the clustering classification. This has shown to enhance the confidence of its users overall on making successful monitoring and diagnosis on particular cases. WIMS was called to explain its provisional judgements in more than one system that both of which were dealing with uncertainty. This can be seen from the conducted experiments, which demonstrated with success the explanation provision at both micro level (individual cases) and macro level (case justification among the case-base overall). Several challenges have been raised while attempting to offer interoperable explanation provision but were dealt successfully with the application of diverse techniques.

The current chapter finalises the research approach towards the intelligent monitoring of workflows by presenting the concept of explanation in CBR-WIMS. The relevant research work was presented as well as the experiments conducted for the evaluation of the adopted research approach. The next chapter concludes the current research approach by summarising the findings and conclusions of this thesis as well as the future work.

Chapter 7

Conclusion and further work

7.1 Summary

This thesis investigated the use of case-based reasoning as a tool for the effective monitoring of business processes. In order to answer the main research questions posed in Chapter 1 (sections 1.2 & 1.3) a number of different approaches were considered towards the intelligent monitoring and diagnosis of business processes. An approach based on temporal logic was chosen and proposed, tested over a number of real-world applications and has shown that CBR can be used effectively to do that.

The literature survey has shown the state of the art in the related disciplines of this research work. However, a unified approach towards the intelligent monitoring and diagnosis of workflows does not exist. Therefore a concrete research approach was formulated (research methods) in order to address the given problem.

Throughout this research several other research disciplines were researched including workflow management, temporal logic and existing time theories, areas of uncertainty and event mining representation of graphs and graph-similarity as well as the concepts of explanation and reasoning provision.

This chapter presents the conducted research work in pursuit of the answers to the questions stated at the introduction, the contributions of this thesis as well as the arisen questions for possible future research work.

As set out in the introduction the principal research question was whether there can be intelligent monitoring and diagnosis on business processes using case-based reasoning. Since business processes are being monitored by human auditors, their explicit monitoring approaches were taken into account. As described throughout the introduction the main aims of this research were:

- To understand what is a suitable representation of a business process
- To understand how similarity measures among workflows can be established
- To define how the explanation of a monitoring process can be presented to its users along with any related insights
- To investigate which software architecture can support a CBR system to effectively monitor business processes

Towards the investigation of the above aims, two workflow-orchestrated enterprise systems were chosen as a test bed for the research overall. The selected systems presented characteristics of modern organisations, using business processes for the coordination and orchestration of their procedures.

As seen from the exhaustive literature research, the execution of business processes can be represented in terms of workflows, giving insights of how tasks and stakeholders interact throughout a pre-designed process. These workflows can afterwards be represented in terms of graphs and be compared by applying similarity measures on them.

Workflows, although efficient in monitoring structured procedures, can state failures while attempting to monitor and interpret effectively the actual state of a business process that includes uncertainty. The human actors of a business process can behave in an unpredictable way that could even reach regions outside its formal specification. Actions without rationale that take place within the execution of a workflow can lead to uncertainty. A system that monitors such workflows should be able to take into account several parameters in order to be able to tackle uncertainty.

For the needs of the evaluation of this research the University of Greenwich, CMS exam moderation system has been used as a case study and several sets of experiments have been conducted. Via those experiments it has been shown that there can be efficient monitoring of a business process by using Case-based reasoning. This has been based on the assumption that the provided cases are accompanied with the necessary past experience given by domain experts. These experiments along with their produced results have been discussed throughout this thesis.

This research investigated whether there can be a generic approach towards the monitoring of business processes. A generic architectural framework has been developed, named CBR-WIMS. The approach of the addressed system has been presented, highlighting how its provided architecture can smoothly incorporate external systems without affecting their structural integrity and/or provided functionality. For the needs of the evaluation of the WIMS generic characteristics a different enterprise business process system was used (BTS).

Experiments conducted with the EMS and the BTS enterprise systems, have shown the interoperability capabilities of the framework. CBR-WIMS can be used for the knowledge acquisition of business processes and reuse them afterwards to similar systems. An example of such knowledge acquisition can be borrowed from everyday life: a system that contains knowledge regarding school timetabling can serve as a knowledge base for any other system that deals with timetabling, provided the necessary adaptations. This is applicable since the base process contains certain sequences of events and encountered patterns that can frequently occur to similar systems. CBR-WIMS via the experiments conducted on the above mentioned systems has shown that such knowledge transferability is applicable and its resilient architecture can support it.

This research can be used towards the reasoning and sufficient explanation provision of the conducted monitoring. This can be done be illuminating insights of the similarity calculations to a human operator/manager as well as providing suggestions based on past available solutions to similar cases. In order to achieve this, data mining is being applied to the most similar cases as derived from the knowledge repository. This thesis has shown how CBR-WIMS can significantly help towards the successful diagnosis of an investigated case.

7.2 Contribution of this Research work

The main contribution of this thesis is the conclusion that there can be efficient monitoring and diagnosis of a business process using case-based reasoning. The outcomes of this research point out that a business process can be represented efficiently in terms of a workflow. Workflows can represent a business process with precision and this information can be imported to specialised systems that deal with its intelligent management.

An additional contribution is the formulation of a novel monitoring platform which is based on an effective business process representation and can provide explicit monitoring and diagnosis to the selected process. This platform provides a generic environment to facilitate a business process in a BPMN or XPDL format in addition to its knowledge repository. The latter can contain any provided past experience regarding the investigated system and serves as a reference case base that can be used for explanation provision.

Deriving contribution of this thesis is the proposition of adequate similarity measures for business processes from different operational domains. The business processes could contain sequences of timed events that could be compared with graph matching. This thesis has shown two major algorithms: the Components one and the Maximum Common Sub-graph as well as an advanced variant of the MCS for the calculation of similarity measures. The results of those were used afterwards towards the successful diagnosis of business process execution instances.

A further contribution is the origination of a primary architecture that supports the effective monitoring of business processes using CBR. This architecture has been rationalised via a generic attitude that consequently offers a blueprint for the monitoring of enterprise schemes. Throughout this research it has been utilised twice in enterprise systems proving its design feasibility and applicability on business processes.

Final contribution of this thesis is the work conducted in the literature of the surrounding areas, giving a concrete review of the involved interdisciplinary fields such as the workflow intelligent management, business processes in general, temporal representation and uncertainty in systems. This review can be regarded as another contribution of this work.

7.3 Future work

Besides the thesis contributions stated above, the future research paths that it motivates can also be regarded as a real contribution. The main future improvements are summarised below as well as a brief discussion upon them.

Regarding the actual business process representation, an interesting perspective is being highlighted regarding the formal workflow representation in terms of time. This is an area that carries a lot of interest and shows signs of maturity nowadays regarding its tangibility; therefore it should be further explored. This as well as the investigation of applying optimised graph similarity measures along workflows can be part of future research.

Part of the future work can be a more in depth investigation of the knowledge interoperability that the developed architecture offers. Future work on this platform could investigate how this knowledge transferability can be exploited efficiently and be of benefit to systems of a larger scale, or systems which may have been recently deployed and might not contain a robust past experience repository. Working with data from business processes of different application areas could investigate up to which extent there can be a safe monitoring generalisation. The reverse research approach could be also investigated: how from a general monitoring approach there can be local optimisation based on the explicit characteristics of a process.

Taking a step ahead but based on the architecture proposed, further work can look into the issue of temporal uncertainty combining contextual temporal information to enhance the reasoning process. Further tests could evaluate the ability of the developed framework to adapt to changing business processes with minimum loss of past useful experience.

There could also be evaluation of the ability to reason across similar but not identical business processes. This knowledge reuse among processes is currently at a primitive stage and should be further expanded.

Finally, the provenance of any used cases and their associated solutions should be investigated further, identifying up to which extent they could assist and augment the explanation process.

Appendix A

A.1 Related publications published in the duration of this thesis

Kapetanakis, S., Petridis, M., Ma, J., Knight, B., Bacon, L. (2011). Enhancing Similarity Measures and Context Provision for the Intelligent Monitoring of Business Processes in CBR-WIMS. In: *Proceedings of PO-CBR: Process-oriented Case-Based Reasoning*, *ICCBR 2011 workshop*. Greenwich, London, UK.

Kapetanakis, S., Petridis, M., Ma, J., Knight, B. (2010). CBR-WIMS, an Intelligent Monitoring Platform for Business Processes. In Petridis, M. (ed.): *Proceedings of the 15th UK CBR workshop*, pp. 55-63. Cambridge: CMS press.

Kapetanakis, S., Petridis, Ma, J., Bacon, L. (2010). Providing Explanations for the Intelligent Monitoring of Business Workflows Using Case-Based Reasoning. In Roth-Berghofer, T., Tintarev, N., Leake, D. B., Bahls, D. (eds.): *Proceedings of the Fifth International workshop on Explanation-aware Computing ExaCt* (ECAI 2010). Lisbon, Portugal.

Kapetanakis, S., Petridis, M., Knight, B., Ma, J., Bacon, L. (2010). A case based reasoning approach for the monitoring of business workflows. In Bichindaritz, I., Montani, S. (eds.): *18th International Conference on Case-Based Reasoning, ICCBR 2010, LNCS (LNAI), vol. 6176,*pp. 390 – 405. Alessandria, Italy: Springer.

Kapetanakis, S., Petridis, M., Ma, J., Bacon, L. (2009). CBR-WIMS, an Architecture for the Intelligent Monitoring and Diagnosis of Workflows. In Petridis, M. (ed.): *Proceedings of the 14th UK CBR workshop*, pp. 67-78. Cambridge: CMS press.

Kapetanakis, S., Petridis, M., Ma, J., Bacon, L. (2009). Workflow Monitoring and Diagnosis Using Case Based Reasoning on Incomplete Temporal Log Data. *Proceedings of the UKDS Workshop at the 8th International Conference on Case Based Reasoning.* Seattle, WA, USA. July 20-23. Berlin: Springer.

Appendix B

B.1 EMS Interview Form

EMS Interview Form *(Please notice that this is information is strictly confidential and it will be used only for educational purposes)*
Date:
Expert's Name:
Expert's Position:
Investigated system:
Case Ranking (class A, B, C):
Comments related to the case:
Any other comments:

B.2 BTS Interview Form

BTS Interview Form *(Please notice that this is information is strictly confidential and it will be used only for educational purposes)*
Date:
Expert's Name:
Expert's Position:
Investigated system:
Case Ranking (class safely stored, damaged, lost and no clear status):
Comments related to the case:
Any other comments:

References

[Van der Aalst, 1998] Van der Aalst, W. M. P. (1998). The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers.* **8 (**1), pp. 21-66.

[Van der Aalst & Hoefstede, 2002] Van der Aalst, W. M. P., Hoefstede, A. (2002). Workflow Patterns: On the Expressive Power of (Petri-net-based) Workflow Languages. In: *Jensen, (ed.) Fourth Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*. Aarhus, Denmark, pp. 28-30.

[Van der Aalst & Hee, 2004] Van der Aalst, W. M. P., & Van Hee, K. M. (2004). *Workflow Management. Models, Methods, and Systems*. Massachusetts: The MIT Press.

[Van der Aalst et al., 2003] Van der Aalst, W. M. P., Ter Kofstede, A. H. M., Kiepuszewski, B., Barros, A. P. (2003). Workflow patterns. *Distributed and Parallel Databases*. **14** (1), pp. 5 – 51.

[Van der Aalst et al., 2007] Van der Aalst, W. M. P., Reijers, H. A., Weijters, A. J. M. M., Van Dongen, B. F.,de Medeiros, A. K. A., Song, M., Verbeek, H. M. W. (2007). Business process mining: An industrial application. *Information Systems.* **32** (5), pp. 713 – 732.

[Aamodt & Plaza, 1994] Aamodt, A and Plaza, E, 1994, Case-based reasoning: foundational issues, methodological variations, and system approaches. *AI Communications.* **7**(1), pp. 39–59.

[Acorn & Walden, 1992] Acorn, T. L. and Walden, S. H. (1992). SMART: Support Management Automated Reasoning Technology for Compaq Customer Service. *Proceedings of the IAAI '92 Conference*, July, San Jose.

[Allen, 1984] Allen, J. (1984). Towards a General Theory of Action and Time. *Artificial Intelligence.* **23** (2), pp. 123-154.

[Allen & Hayes, 1989] Allen, J., Hayes, P. (1989). Moments and Points in an Interval-based Temporal-based Logic. *Computational Intelligence.* **5** (4), pp. 225-238.

[Arditi & Tokdemir, 1999] Arditi, D., Tokdemir, O. B. (1999). Comparison of Case-Based Reasoning and Artificial Neural Networks. *Journal of Computing in Civil Engineering*. **13** (3), pp. 162 – 169.

[Ashley & Rissland, 1987] Ashley, K. D., Rissland, E. L. (1987). Compare and contrast: A test experience. In *Proceedings AAAI'87*. Seattle, WA, pp. 273-278.

[Atzmueller & Roth-Berghofer,2010] Atzmueller, M. & Roth-Berghofer, T. (2010). Towards Explanation-Aware Social Software: Applying the Mining and Analysis Continuum of Explaining. In Roth-Berghofer, T., Tintarev, N., Leake, D. B., Bahls, D. (eds.)*: Proceedings of the Fifth International workshop on Explanation-aware Computing ExaCt* (ECAI 2010). Lisbon, Portugal.

[Badra et al., 2009] Badra, F., Cojan, J., Cordier, A., Lieber, J., Meilender, T., Mille, A., Molli, P., Nauer, E., Napoli, A., Skaf-Molli, H., Toussaint, Y. (2009). Knowledge acquisition and discovery for the textual case-based cooking system WikiTaaable. In: Sarah Jane Delany (ed.), *8th International Conference on Case-Based Reasoning - ICCBR 2009, Workshop Proceedings*, Seattle, WA, USA, pp. 249–258.

[Baldan et al., 2005] Baldan, P., Corradini, A., Ehrig, H., Heckel, R. (2005). Compositional semantics for open Petri nets based on deterministic processes. *Mathematical Structures in Computer Science*. **15** (1), pp. 1-35.

[Bandara et al., 2007] Bandara, W., Indulska, M., Chong, S., Sadiq, S. (2007) Major Issues in Business Process Management: An Expert Perspective. In: *ECIS 2007 - The 15th European Conference on Information Systems*. St Gallen, Switzerland, pp. 1240 – 1251.

[Bates, 1993] Bates, P. (1993). *Workflow Technology A Discussion Paper*. [Online]. Available at: http://www.batescommunications.net/WORKFLOW.pdf/. [Accessed 20 December 2011]

[van Benthem, 1983] van Benthem, J. (1983). *The Logic of Time*. Dordrech: Kluwer Academic.

[Bergmann, 2002] Bergmann, R. (2002). *Experience Management - Foundations, Development Methodology, and Internet-Based Applications*. Berlin: Springer-Verlag.

[Berndt & Clifford, 1996] Berndt, D. J., Clifford, J. (1996). Finding patterns in time series: a dynamic programming approach. In: *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, Menlo Park, CA, pp. 229-248.

[Biocare Medical, 2009] Biocare Medical. (2009). *intelliPATH Automated Staining Instrument.* [Online]. Available at: http://www.slidestainer.com/. [Accessed 3 November 2011].

[Bonissone & Cheetham, 1998] Bonissone, P., Cheetham, W. (1998). Fuzzy Case-Based Reasoning for Residential Property Valuation. In: Ruspini, Bonissone, Pedrycz (Eds.) *Handbook of Fuzzy Computing.* Oxford: Oxford University Press, Section G14.1.

[Bruce, 1972] Bruce, B. (1972). A Model for Temporal References and Application in a Question Answering Program, *Artificial Intelligence*, **3**, pp. 1 - 25.

[Bunke, 2000] Bunke, H. (2000). Graph Matching: Theoretical Foundations, Algorithms, and Applications. In: *International Conference on Vision Interface*. Montreal, Canada, pp. 82 - 88.

[Carbonell, 1983] Carbonell, J. (1983). Learning by analogy: Formulating and generalizing plans from past experience. In: Michalski, Carbonel & Mitchell (eds.). *Machine Learning: An Artificial Intelligence Approach.* Tioga, Palo Alto, pp. 137-162.

[Cheetham, 2005] Cheetham, W. (2005). Tenth Anniversary of Plastics Color Matching. *Artificial Intelligence Magazine*. **26** (3), pp. 51-61.

[Cheetham & Goebel, 2007] Cheetham, W., Goebel, K. (2007). Appliance Call Center: A Successful Mixed-Initiative Case Study. *Artificial Intelligence Magazine*. **28** (2), pp. 89-100.

[Chen et al., 2000] Chen, Q., Dayal, U., Hsu, M., Griss, M. (2000). Dynamic-agent, Workflow and XML for E-Commerce Automation. Lecture Notes in Computer Science, pp. 314 -323.

[Chinthaka et al., 2009] Chinthaka, E., Ekanayake, J., Leake, D., Plale, B. (2009). CBR Based Workflow Composition Assistant. In *Proceedings of Services 2009*. Washington, DC: IEEE Computer Society, pp. 352 – 355.

[Clark, 1952] Clark, W. (1952). *The Gantt Chart* (3rd edition). London: Pitman.

[CNET, 2009] CNET. (2009). Police can't cope with volume of CCTV data [Online]. Available at: http://crave.cnet.co.uk/software/police-cant-cope-with-volume-of-cctv-data-49302282/. [Accessed 5 May 2012].

[Cohnitz, 2000] Cohnitz., D. (2000). Explanations are like salted peanuts: On Why You Can't Cut the Route toward Further Reduction. In: *Proceedings of the Fourth International Congress of the Society for Analytic Philosophy*, pp. 22–36.

[Collins, 1987] Collins, G. (1987). *Plan creation: Using strategies as blueprints*. PhD thesis. Yale University.

[Cordier et al., 2009] Cordier, A., Lieber, J., Molli, P., Nauer, E., Skaf-Molli, H., Toussaint, Y. (2009). WikiTaaable: A semantic wiki as a blackboard for a textual case-based reasoning system. In: *Proceedings of SemWiki 2009 – 4th Semantic Wiki Workshop at the 6th European Semantic Web Conference - ESWC 2009*. Heraklion, Greece.

[Cram et al., 2008] Cram, D., Fuchs, B., Prié, Y. & Mille, A. (2008). An approach to User-Centric Context-Aware Assistance based on Interaction Traces. *Fifth International Workshop Modeling and Reasoning in Context (MRC 2008)*. Delft, The Netherlands.

[Cram, Mathern & Mille, 2011] Cram, D., Mathern, B., Mille, A. (2011). A Complete Chronicle Discovery Approach: Application to Activity Analysis. *Expert Systems, The Journal of Knowledge Engineering*.

[Crowley, 2010] Crowley, J. L. (2010). *Pattern Recognition and Machine Learning*. [Online]. Available at: http://www-prima.imag.fr/jlc/Courses/2010/ENSI3.PRML/ENSI2.PRLM.S2.pdf . [Accessed 20 July 2011].

[CSML, 2010] CSML (2010). *Switching mechanism of lambda phage*. [Online]. Available at: http://www.csml.org/models/csml-models/switching-mechanism-of-lambda-phage. [Accessed 10 November 2010].

[Cunningham et al., 2004] Cunningham, C., Weber, R., Proctor, J. M., Fowler, C., Murphy, M. (2004). Investigating Graphs in Textual Case-Based Reasoning. In: Funk P., González Calero P. A. (eds.*). Advances in Case-Based Reasoning: Proceedings of the Seventh European Conference (ECCBR 2004), LNAI 3155*, pp. 573- 586. Berlin: Springer.

[Davenport & Short, 1990] Davenport, T., Short, J. (1990). The New Industrial Engineering: Information Technology and Business Process Redesign. In: *Sloan Management Review*, Summer 1990, pp 11–27.

[Davenport, 1993] Davenport, T. (1993). *Process Innovation: Reengineering work through information technology*. Boston: Harvard Business School Press.

[De Miguel, Plaza & Díaz-Agudo, 2008] De Miguel, J. Plaza, L., Díaz-Agudo, B. (2008). Colibricook: A cbr system for ontology based recipe retrieval and adaptation. In: *Proceedings of ECCBR Workshops 2008*, pp. 199-208.

[Díaz-Agudo et al., 2007] Díaz-Agudo, B., González-Calero, P., Recio-García, J., Sánchez, A. (2007). Building cbr systems with jcolibri. *Special Issue on Experimental Software and Toolkits of the Journal Science of Computer Programming.* **69** (1-3), pp. 68-75.

[Dufour-Lussier et al., 2010] Dufour-Lussier, V., Lieber, J., Nauer, E., Toussaint, Y. (2010). Text adaptation using formal concept analysis. In: Bichindaritz, I., Montani, S. (eds.). *ICCBR 2010, Lecture Notes in Computer Science*. 6176, pp.96—110.

[Dousson, Gaborit & Ghallab, 1993] Dousson, C., Gaborit, P., Ghallab, M. (1993). Situation recognition: Representation and algorithms. In: *Proceedings of IJCAI 1993*. Chambéry, France, pp. 166 - 174.

[East West Bank, 2012] East West Bank. (2012). *Personal Banking - Loans & Lines of Credit*. [Online]. Available at: http://www.eastwestbank.com/english/PB_HLProcess.asp. [Accessed 3 April 2012].

[Van Elst et al., 2003] Van Elst, L., Aschoff, F. R., Bernardi, A., Maus, H., Schwarz, S. (2003). Weakly-structured workflows for knowledge-intensive tasks: An experimental evaluation. *12th IEEE International Workshops on Enabling Technologies Infrastructure for Collaborative Enterprises* (WETICE 2003). Linz, Austria: IEEE Computer Society, pp. 340 – 345.

[EUT/D&T Bakkenist, 2000] EUT/D&T Bakkenist. (2000). *ExSpect*. [Online]. Available at: http://www.exspect.com/. [Accessed 2 December 2011].

[Fakas & Karakostas, 1999] Fakas, G., Karakostas, B. (1999). A workflow management system based on intelligent collaborative objects. *Information and Software Technology.* **41** (13), pp. 907–915.

[Feldman, Gupta & Srivastava, 2010] Feldman, S., Gupta, M.R., Srivastava, S. (2010). Completely Lazy Learning. *Knowledge and Data Engineering, IEEE Transactions on*. **22** (9), pp. 1274-1285.

[Fisher & Sanderson, 1996] Fisher, C., Sanderson, P. (1996). Exploratory sequential data analysis: exploring continuous observational data. *Interactions.* **3** (2), pp. 25 - 34.

[Floyd & Esfandiari, 2011] Floyd, M. W., Esfandiari, B. (2011). Learning State-Based Behaviour using Temporally Related Cases. In: *Proceedings of the 16th United Kingdom Workshop on Case-Based Reasoning*. Cambridge, England, UK, pp. 34-45.

[Freßmann et al., 2005] Freßmann, A., Maximini, R., Sauer, T. (2005). Towards Collaborative Agent-based Knowledge Support for Time-critical and Business-critical Processes. *Professional Knowledge Management: Third Biennial Conference, WM*. Kaiserslautern, Germany: Springer – Verlag, pp. 420 – 430.

[Galton, 1990] Galton, A. (1990). A Critical Examination of Allen's Theory of Action and Time. *Artificial Intelligence.* **42** (2), pp.159-188

[Georgakopoulos et al., 1995] Georgakopoulos, D., Hornick, M., Sheth, A. (1995). An Overview of Workflow Management. *Distributed and Parallel Databases*. **3** (2), pp. 119 – 153.

[Georgeon, Mille & Bellet, 2006] Georgeon, O., Mille, A. & Bellet, T. (2006). Analysing behavioural data for refining cognitive models of operator. In Society (ed.): *Philosophies and Methodologies for Knowledge Discovery. Seventeenth international Workshop on Database and Expert Systems Applications.* Krakow, Poland: IEEE Computer Society Press.

[Goel, 1989] Goel, A. (1989). *Integration of case-based reasoning and model-based reasoning for adaptive design problem solving.* Ph.D. Thesis. The Ohio State University.

[Goel & Chandrasekaran, 1989] Goel, A., Chandrasekaran, B. (1989). Use of device models in adaptation of design cases. In Hammond (ed.): *Proceedings Second Workshop on case-based reasoning*. Pensacola Beach, Florida: Morgan-Kauffman.

[Halpern & Shoham, 1991] Halpern, J., Shoham, Y. (1991). A Propositional Model Logic of Time Intervals, *Journal of the Association for Computing Machinery*. **38** (4), pp. 935-962.

[Hammer & Champy, 1993] Hammer, M., Champy, J. (1993). Reengineering the Corporation: A Manifesto for Business Revolution. Harper Business Books, New York.

[Hammond, 1986] Hammond, K. J. (1986). CHEF: A model of case-based planning. In: *Proceedings of AAAI-86*. California, US: Morgan-Kaufmann, pp. 267-271.

[Hammond, 1987] Hammond, K. J. (1987). Explaining and repairing plans that fail. In: *Proceedings International Joint Conference on Artificial Intelligence, IJCAI-87*, pp. 109 - 114.

[Hammond, 1989] Hammond, K. J. (1989). Case-based planning: Viewing planning as a memory task. *Perspectives in Artificial Intelligence*. San Diego, CA, USA: Academic Press.

[Hanft, Ihle & Newo, 2009] Hanft, A., Ihle, N., Newo, R. (2009). Refinements for Retrieval and Adaptation of the CookIIS Application. Wissensmanagement, pp. 139 - 153.

[Hare, 2010] Hare, T. (2010). *Nearest Neighbour bases approaches to Multivariate Data Analysis* [Online]. Available at: http://www.slideshare.net/tim_hare/agglomerative-hierarchical-amp-divisive-clustering. [Accessed 21 May 2011].

[Hassin, Norwawi & Aziz, 2006] Hassin, M. H. B. M., Norwawi, N. B. M., Aziz, A. (2006). Temporal Case-Based Reasoning for reservoir spillway gate operation recommendation. In *Proceedings of the IEEE International Conference in Computing and Informatics (ICOCI'06)*. Kuala Lumpur, pp.1-4.

[Hilbert & Redmiles, 2000] Hilbert, D. M. & Redmiles, D. F. (2000). Extracting usability information from user interface events. *ACM Computing Surveys* **32** (4), pp. 384 - 421.

[Hinkle & Toomey, 1994] Hinkle, D., Toomey, C. N. (1994). Applying case-based reasoning to composite part fabrication. In: *Proceeding of the Sixth Innovative Application of AI Conference*. Seattle, WA: AAAI Press, pp. 55-62.

[Hinrichs, 1988] Hinrichs, T. R. (1988). Towards an architecture for open world problem solving. In Kolodner (ed.): *Proceedings Case-Based Reasoning Workshop*. San Mateo, California: Morgan-Kauffman.

[Hinrichs, 1989] Hinrichs, T. R. (1989). Strategies for adaptation and recovery in a design problem solver. In Hammond (ed.): *Proceedings Second Workshop on case-based reasoning*. Pensacola Beach, Florida: Morgan-Kauffman.

[Hofstadter, 1985] Hofstadter, D. R. (1985). Analogies and roles in human and machine thinking. In: *Metamagical Themas*. New York: Basic Books, pp. 547-603.

[IBM, 2003] IBM (2003). *Business Process Execution Language for Web Services.* [Online]. Available at: http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel/ws-bpel.pdf. [Accessed 17 March 2011].

[IBM, 2007] IBM (2007). *Business process standards, Part 1: An introduction.* [Online]. Available at: http://www.ibm.com/developerworks/websphere/library/techarticles/0710_fasbinder/0710_fasbinder.html. [Accessed 1 March 2011].

[IBM & BEA Systems, 2004] IBM & BEA Systems (2004). *BPELJ: BPEL for Java technology.* [Online]. Available at: https://www.ibm.com/developerworks/library/specification/ws-bpelj. [Accessed 7 June 2011].

[IBM & SAP, 2005] IBM & SAP (2005). *WS-BPEL Extension for People – BPEL4People.* [Online]. Available at: http://public.dhe.ibm.com/software/dw/specs/ws-bpel4people/BPEL4People_white_paper.pdf. [Accessed 19 March 2011].

[Jære, Aamodt & Skalle, 2002] Jære, M. D., Aamodt, A., Skalle, P. (2002). Representing Temporal Knowledge for Case-Based Prediction. In Craw, S., Preece, A.D. (eds.): *Proceeding of ECCBR 2002. LNCS (LNAI), vol. 2416*, pp. 174 – 188. Heidelberg: Springer.

[Jain, Murty & Flynn, 1999] Jain, A. K., Murty, M. N., Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Survey*, **31** (3), pp. 264-323.

[Jennings et al., 2000] Jennings, N. R., Faratin, P., Norman, T. J., O'Brien, P., Odgers, B. (2000). Autonomous Agents for Business Process Management. *Int. Journal of Applied Artificial Intelligence*, **14** (2), pp. 145 - 189.

[Jha, 2007] Jha, G., K. (2007). Artificial Neural Networks and its Applications. [Online]. Available at: http://www.iasri.res.in/ebook/EB_SMAR/e-book_pdf%20files/Manual%20IV/3-ANN.pdf. [Accessed 7 May 2012].

[Judge et al., 1998] Judge, D. W., Odgers, B. R., Shepherdson, J. W., and Cui Z. (1998). Agent Enhanced Workflow. *BT Technical Journal*. **16** (3), pp. 79 - 85.

[Jung et al., 2004] Jung, J. Y., Hur, W., Kang, S. H., Kim, H. (2004). Business Process Choreography for B2B Collaboration. *IEEE Internet Computing*. **8** (1), pp. 37-45.

[Kapetanakis et al., 2009a] Kapetanakis, S., Petridis, M., Ma, J., Bacon, L. (2009). Workflow Monitoring and Diagnosis Using Case Based Reasoning on Incomplete Temporal Log Data. *Proceedings of the UKDS Workshop at the 8th International Conference on Case Based Reasoning.* Seattle, WA, USA. July 20-23. Berlin: Springer.

[Kapetanakis et al., 2009b] Kapetanakis, S., Petridis, M., Ma, J., Bacon, L. (2009). CBR-WIMS, an Architecture for the Intelligent Monitoring and Diagnosis of Workflows. In Petridis, M. (ed.): *Proceedings of the 14th UK CBR workshop*, pp. 67-78. Cambridge: CMS press.

[Kapetanakis et al., 2010a] Kapetanakis, S., Petridis, M., Knight, B., Ma, J., Bacon, L. (2010). A case based reasoning approach for the monitoring of business workflows. In Bichindaritz, I., Montani, S. (eds.): *18th International Conference on Case-Based Reasoning, ICCBR 2010, LNCS (LNAI), vol. 6176,*pp. 390 – 405. Alessandria, Italy: Springer.

[Kapetanakis et al., 2010b] Kapetanakis, S., Petridis, Ma, J., Bacon, L. (2010). Providing Explanations for the Intelligent Monitoring of Business Workflows Using Case-Based Reasoning. In Roth-Berghofer, T., Tintarev, N., Leake, D. B., Bahls, D. (eds.*): Proceedings of the Fifth International workshop on Explanation-aware Computing ExaCt* (ECAI 2010). Lisbon, Portugal.

[Kapetanakis et al., 2010c] Kapetanakis, S., Petridis, M., Ma, J., Knight, B. (2010). CBR-WIMS, an Intelligent Monitoring Platform for Business Processes. In Petridis, M. (ed.): *Proceedings of the 15th UK CBR workshop*, pp. 55-63. Cambridge: CMS press.

[Kapetanakis et al., 2011] Kapetanakis, S., Petridis, M., Ma, J., Knight, B., Bacon, L. (2011). Enhancing Similarity Measures and Context Provision for the Intelligent Monitoring of Business Processes in CBR-WIMS. In: *Proceedings of PO-CBR: Process-oriented Case-Based Reasoning*, *ICCBR 2011 workshop*. Greenwich, London, UK.

[Kappel et al., 1995] Kappel, G., Lang, P., Rausch-Schott, S., Retschitzegger, W. (1995). Workflow Management Based on Objects, Rules, and Roles. *IEEE Bulletin of the Technical Committee on Data Engineering*. **18** (1), pp. 11 – 18.

[Kersten & Meister, 1996] Kersten, G. E., Meister, D. B. (1996). Qualitative representations of negotiations: Tutorial on a rule-based approach. *Group Decision and Negotiation*. **5** (2), pp. 177 – 202.

[Kim et al., 2002] Kim, J., Suh, W., Lee, H. (2002). Document – based workflow modelling: a case – based reasoning approach. *Expert System with Applications*. **23** (1), pp. 77 – 93.

[Kolodner, 1983] Kolodner, J. L. (1983). Reconstructive memory: A computer model. *Cognitive Science*. **7** (4), pp. 281-328.

[Kolodner, 1993] Kolodner, J. (1993). *Case-based reasoning*. San Mateo, California: Morgan Kaufmann

[Koton, 1989] Koton, P. (1989). *Using experience in learning and problem solving*. (PhD thesis). Massachusetts Institute of Technology, Laboratory of Computer Science.

[Kueng, 2000] Kueng, P. (2000). The Effects of Workflow Systems on Organizations: A Qualitative Study. In: *proceedings of Business Process Management, Models, Techniques, and Empirical Studies*. Berlin: Springer-Verlag, pp. 301-316.

[Kumar & Wang, 2010] Kumar, A, Wang, J. (2010). A framework for designing resource driven workflow systems. In: Rosemann M., vom Brocke J. (eds.). *The International Handbook on Business Process Management 1.* Berlin : Springer-Verlag, pp. 419 – 440.

[Ladkin, 1992] Ladkin, P. (1992), Effective solutions of qualitative intervals constraint problems, *Artificial Intelligence*, **57** (1), pp. 105-124.

[Leake, 2002] Leake, D. B. (2002). Artificial Intelligence (A brief non-technical introduction). In V*an Nostrand's Scientific Encyclopedia*. Vol. 1. Ninth Edition. New York: Wiley.

[Leake & Whitehead, 2007] Leake, D. B., Whitehead, M. (2007). Case provenance: The value of remembering case sources. In: *Proceedings of the 7th international conference on Case-Based Reasoning: Case-Based Reasoning Research and Development (ICCBR-07*). Berlin: Springer-Verlag.

[Leake, 2008] Leake, D., B. (2008). Provenance and Case-Based Reasoning. In Wilson, D., Lane, H. C. (eds.): Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference (FLAIRS), Coconut Grove, Florida, USA: AAAI Press.

[Lebowitz, 1983] Lebowitz, M. (1983). Memory-based parsing. *Artificial Intelligence*. **21** (4), pp. 363-404.

[Leymann & Altenhuber, 1994] Leymann, F., Altenhuber,W. (1994). Managing Business Processes as an Information Resource, *IBM Systems Journal.* **33** (2), pp. 326 – 348.

[Likhachev, Kaess & Arkin, 2002] Likhachev, M., Kaess, M., Arkin, R. C. (2002). Learning behavioral parameterization using spatio-temporal case-based reasoning. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. **2**, pp. 1282-1289.

[Liu, Li & Zhao, 2009] Liu, C., Li, Q., Zhao, X. (2009). Challenges and opportunities in collaborative business process management: Overview of recent advances and introduction to the special issue. *Information Systems Frontiers.***11** (3), pp. 201 – 209.

[Ma, 2007] Ma, J. (2007). Ontological considerations of time, meta-predicates and temporal propositions. *Applied Ontology.* **2** (1), pp. 37 - 66.

[Ma & Knight, 1994] Ma, J., Knight, B. (1994). A General Temporal Theory. *The Computer Journal.* **37** (2), pp. 114-123.

[Ma & Knight, 2001] Ma. J., Knight, B. (2001). Reified Temporal Logics: An Overview. *Artificial Intelligence Review*. **15** (3), pp. 189 – 217.

[Ma & Knight, 2003] Ma, J. Knight, B. (2003). Representing the Dividing Instant. *The Computer Journal.* **46** (2), pp. 213-222.

[Ma, Knight & Petridis, 2008] Ma, J., Knight, B., Petridis, M. (2008). Deriving Explanations from Partial Temporal Information. In: *workshop proceedings of ExACT-08* (ECAI 2008). Patras, Greece.

[Mantaras et al., 2005] Mantaras, R., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M., Cox, M., Forbus, K., Keane, M., Aamodt, A., and Watson, I. (2005). Retrieval, reuse, revision and retention in CBR. *Knowledge Engineering Review*. **20** (3), pp. 215 - 240.

[Marcella & Middleton, 1996] Marcella, R., Middleton, I. (1996). The Role of the Help Desk in the Strategic Management of Information Systems. *OCLC Systems and Services.* **12** (4), pp. 4-19.

[McDermott, 1982] McDermott, D. (1982), A Temporal Logic for Reasoning about Processes and Plans, *Cognitive Science*, **6** , pp. 101-155.

[Mentzas et al, 2001] Mentzas, G., Halaris, C., Kavadias, S. (2001). Modelling business processes with workflow systems: an evaluation of alternative approaches. *International Journal of Information Management*. **21** (2), pp. 123-135.

[Michaelis et al, 2009] Michaelis, J. R., Ding, L., McGuinness, D. L. (2009). Towards the Explanation of Workflows. In: *Proceedings of the IJCAI 2009 Workshop on Explanation Aware Computing (ExaCt)*. Pasadena, CA, US.

[Mileman et al., 2002] Mileman, T., Knight, B., Petridis, M., Cowell, D., Ewer, J. (2002). Case-Based Retrieval of 3-D shapes for the design of metal castings. *Journal of Intelligent Manufacturing*. **13** (1), pp. 39-45.

[Minor et al., 2007a] Minor, M., Tartakovski, A., Bergmann, R. (2007). Representation and Structure-Based Similarity Assessment for Agile Workflows. *Proceedings of the 7th international conference on Case-Based Reasoning* (ICCBR 2007). Belfast, NI, UK: Springer – Verlag, pp. 224 – 238.

[Minor et al., 2007b] Minor, M., Schmalen, D., Koldehoff, A., Bergmann, R. (2007). Structural Adaptation of Workflows Supported by a Suspension Mechanism stand by Case-Based Reasoning. *16th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises* (WETICE 2007). Paris, France : IEEE Computer Society, pp. 370-375

[Minor, Bergmann & Görg, 2011] Minor, M., Bergmann, R., Görg, S. (2011). Adaptive Workflow Management in the Cloud – Towards a Novel Platform as a Service, In: *Proceedings of the Process-oriented Case-Based Reasoning workshop*. International Conference on Case-Based Reasoning (ICCBR), London.

[Mitchell, 1993] Mitchell, M. (1993). *Analogy-Making as Perception: A Computer Model*. Cambridge, Massachusetts, U.S.A: Bradford Books.

[Montani & Leonardi, 2010] Montani, S., Leonardi, G. (2010). A Case-Based Approach to Business Process Monitoring. In: Bramer, M. (ed.): IFIP AI 2010. IFIP AICT, **331**, pp. 101–110. Springer, Heidelberg.

[Montani, Leonardi & Lo Vetere, 2011] Montani, S., Leonardi, G., Lo Vetere, M. (2011). Case Retrieval and Clustering for Business Process Monitoring. In: *Proceedings of the Process-oriented Case-Based Reasoning workshop*. International Conference on Case-Based Reasoning (ICCBR), London.

[Montani & Leonardi, 2012] Montani, S., Leonardi, G. (2012). Retrieval and clustering for business process monitoring: results and improvements. In: Diaz Agudo, B., Watson, I. (eds.): *Proceedings of the International Conference on Case Based Reasoning* (ICCBR) 2012, LNCS 7466, pp. 269–283, Springer, Berlin.

[Navarro et al., 2012] Navarro, M., De Paz, J. F., Julián, V., Rodríguez, S., Bajo, J., Corchado, J. M. (2012). Temporal bounded reasoning in a dynamic case based planning agent for industrial environments. *Expert Systems with Applications*. **39** (9), pp. 7887-7894.

[Navinchandra, 1988] Navinchandra, D. (1988). Case-based reasoning in CYCLOPS, a design problem solver. In Kolodner, J. (ed.): *In Proceedings of the DARPA Workshop on Case-Based Reasoning,* pp. 286–301. San Mateo, California: Morgan-Kauffman.

[Netten, 1998] Netten, B.D. (1998). Representation of Failure Context for Diagnosis of technical Applications. In Smyth B., Cunningham P. (eds.): Advances in Case-Based Reasoning, 4th European Workshop, EWCBR-98, LNAI 1488, pp. 239 – 250. Dublin, Ireland: Springer.

[O'Brien & Wiegand, 1998] O'Brien P.D., Wiegand., W.E. (1998). "Agent based process management: applying intelligent agents to workflow". *The Knowledge Engineering Review*. **13** (2), pp.1-14.

[OASIS, 2006] OASIS (2006). *Reference Model for Service Oriented Architecture 1.0.* [Online]. Available at: http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf. [Accessed 4 January 2011].

[OASIS, 2007a] OASIS (2007a). *Web Services Business Process Execution Language Version 2.0.* [Online]. Available at: http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf. [Accessed 17 March 2011].

[OASIS, 2007b] OASIS (2007b). *Web Services Business Activity (WS-BusinessActivity) Version 1.1.* [Online]. Available at: http://docs.oasis-open.org/ws-tx/wstx-wsba-1.1-spec-os.pdf. [Accessed 2 May 2011].

[OASIS, 2011] OASIS (2011). Service Component Architecture Assembly Model Specification Version 1.1. [Online] Available at: http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-spec.pdf. [Accessed 30 July 2011].

[Odgers et al., 1999] Odgers, B. R., Thompson, S. G., Shepherdson, J., Cui, D., Judge, D., O'Brien, P. D. (1999). Technologies for intelligent workflows: Experiences and lessons. In: *Proceedings of Agent-Based Systems in the Business Context, AAAI 1999 Workshop*. Orlando, Florida, USA: AAAI Press / The MIT Press

[OMG, 2000] OMG. (2000). Workflow Management Facility Specification, V1.2. [Online]. Available at: http://www.omg.org/spec/WfMF/1.2/PDF/. [Accessed 4 November 2010]

[OMG, 2011] OMG. (2011). Business Process Model and Notation (BPMN) Version 2.0. [Online]. Available at: http://www.omg.org/spec/BPMN/2.0. [Accessed 3 June 2011].

[Ong, 2004] Ong, M., Ren, X., Allan, G., Kadirkamanathan, V., Thompson, H. A., Fleming, P. J. (2004). *Workflow Advisor on The Grid, Proc UK e-Science All Hands Meeting, AHM 2004*, pp 864-869. Nottingham, UK.

[Petri, 1962] Petri, C.A. (1962). *Kommunikation mit Automaten.* (PhD thesis). Institut für instrumentelle Mathematik, Bonn.

[Petridis, Saeed & Knight, 2010] Petridis, M., Saeed, S., Knight, B. (2010). An automatic Case Based Reasoning System Using Similarity Measures between 3D Shapes to Assist in the Design of Metal Castings. *Expert Update*. **10** (2), pp. 43 – 51.

[Petridis, Ma & Knight, 2011] Petridis, M., Ma, J., Knight, B. (2011). Temporal model for business process. *Intelligent Decision Technologies*. **5** (4), pp. 321 – 332.

[Phillips, 2004] Phillips, M. (2004). *CMMI V1.1 And Appraisal Tutorial*. Software Engineering Institute, Carnegie Mellon University [Online]. Available at: http://www.sei.cmu.edu/cmmi/presentations/euro-sepg-tutorial/sld001.htm. [Accessed 21 November 2008].

[Prentzas & Hatzilygeroudis, 2007] Prentzas, J., Hatzilygeroudis, I. (2007). Categorizing approaches combining rule-based and case-based reasoning. *Expert Systems*. **24** (2), pp. 97-122.

[Prud'hommeaux & Seaborne, 2008] Prud'hommeaux, E., Seaborne, A. (2008). *SPARQL query language for RDF*. [Online]. Available at: http://www.w3.org/TR/rdf-sparql-XMLres/. [Accessed 21 May 2011].

[Recio-García et al., 2006] Recio-García, J., Díaz-Agudo, B., González-Calero, P., Sánchez-Ruiz-Granados, A. (2006). Ontology based CBR with jCOLIBRI. In: *Applications and Innovations in Intelligent Systems XIV. Proceedings of AI-2006*, Springer , pp. 149-162

[Reichert & Dadam, 1998] Reichert, M., Dadam, P. (1998). ADEPTflex - supporting dynamic changes of workflows without losing control. *Journal of Intelligent Information Systems.***10** (2), pp. 93 – 129.

[Reichert et al., 2003] Reichert, M., Rinderle, S., Dadam, P. (2003). Adept workflow management system: Flexible support for enterprise-wide business processes (tool presentation). In: Van der Aalst, W.M.P. et al. (eds.). *International Conference on Business Process Management* (BPM 2003). Eindhoven, Netherlands. Berlin: Springer – Verlag, pp. 370 – 379.

[Richter, 1995] Richter, M. M. (1995). The knowledge contained in similarity measures. Invited Talk at *the First International Conference on Case-Based Reasoning, (ICCBR'95)*, Sesimbra, Portugal.

[Rissland, 1983] Rissland, E.L. (1983). Examples in legal reasoning: legal hypotheticals. *In: Proceedings Eighth International Joint Conference on Artificial Intelligence (IJCAI-83)*, pp. 90-93, Karlsruhe, Germany.

[Roth-Berghofer, 2003] Roth-Berghofer, T., R. (2003). Knowledge Maintenance of Case-Based Reasoning Systems – The SIAM Methodology. *Zeitschrift KI - Künstlichen Intelligenz*. **17** (1), pp. 55 – 57.

[Roth-Berghofer, 2004] Roth-Berghofer, T., R. (2004). Explanations and Case-Based Reasoning: Foundational Issues. In: Funk P., González Calero P. A. (eds.). *Advances in Case-Based Reasoning*, ECCBR 2004, pp. 389 – 403.

[Rozenfeld, 2007] Rozenfeld, J. (2007). Analysis of Large Data Volumes: Challenges and Solutions. [Online]. Available at: http://www.information-management.com/infodirect/20071012/1093614-1.html?pg=3. [Accessed 20 May 2012].

[Schank, 1983] Schank, R. (1983). *Dynamic Memory: A theory of learning in computers and people*. New York, NY, USA: Cambridge University Press.

[Schank , 1986] Schank, R., C. (1986). Explanation Patterns: Understanding Mechanically and Creatively. Hillsdale, NJ.: Lawrence Erlbaum Associates.

[Schurz, 1995] Schurz, G. (1995). Scientific Explanation: A Critical Survey. *Foundation of Science.* **3**, pp. 429-465.

[Scholarpedia, 2008] Scholarpedia (2008). Petri net. [Online]. Available at: http://www.scholarpedia.org/article/Petri_net. [Accessed 27 May 2011].

[Shoham, 1987] Shoham, Y. (1987). Temporal Logics in AI: Semantical and Ontological Considerations. *Artificial Intelligence*. **33**, pp. 89 - 104.

[Silva Filho et al., 1999] Silva Filho, R. S., Wainer, J., Madeira, E. R. M., Ellis, C. (1999). CORBA Based Architecture for Large Scale Workflow. In: *Proceedings of the 4th International Symposium on Autonomous Decentralized Systems (ISADS '99).* Washington, DC, USA, pp. 276-283.

[Simpson, 1985] Simpson, R. L. (1985). A Computer Model of Case-Based Reasoning in Problem Solving: An Investigation in the Domain of Dispute Mediation. *Technical Report GIT-ICS-85/18. Georgia Institute of Technology, School of Information and Computer Science*. Atlanta, US.

[Skalle & Aamodt, 2004] Skalle, P., Aamodt, A. (2004). Knowledge-based decision support in oil well drilling. In: Zhongzhi Shi (ed.), *Proceedings of the ICIIP 2004, International Conference on Intelligent Information Systems*. Beijing, China.

[Smith, 1910] Smith, A. (1910). *Wealth of Nations*. London: Dent & Sons.

[Smith, 1977] Smith, A. (1977). *An Inquiry into the Nature and Causes of the Wealth of Nations*. Chicago: University of Chicago Press.

[Sørmo et al., 2005] Sørmo, F., Cassens, J., Aamodt, A. (2005). Explanation in case-based reasoning – perspectives and goals. *Artificial Intelligence Review*. **24** (2), pp. 109 – 143.

[Stark & Lachal, 1995] Stark, H., Lachal, L. (1995). *Ovum Evaluates: Workflow*. London, UK: Ovum.

[Swartout, 1983] Swartout, W. R. (1983). XPLAIN: A system for creating and explaining expert consulting programs. *Artificial Intelligence*. **21** (3), pp. 285 – 325.

[The Open Group, 1999] The Open Group. (1999). *Object Management: Common Object Request Broker Architecture (CORBA)*. [Online]. Available at: http://www.opengroup.org/branding/prodstds/x98or.htm. [Accessed: 7 November 2010].

[Tsai et al., 2010] Tsai, Y. L., You C. F., Lin, J. Y., Liu, K. Y. (2010). Knowledge-based engineering for process planning and die design for automotive panels. *Computer-Aided Design and Applications*. **7** (1), pp. 75 - 87.

[University of Aarhus, 2006] University of Aarhus. (2006). *Design/CPN - Computer Tool for Coloured Petri Nets*. [Online]. Available at: http://www.daimi.au.dk/designCPN/. [Accessed: 2 December 2011].

[Varma, Aggour & Bonissone, 2005] Varma, A., Aggour, K., S., Bonissone, P. P. (2005). Selecting the Best Units in a Fleet: Performance Prediction from Equipment Peers. In: *Proceedings of the 7th Int. Conference on Case-Based Reasoning (ICCBR2005)*, pp. 595-609.

[Varma & Roddy, 1999] Varma, A., Roddy, N. (1999). ICARUS: design and deployment of a case-based reasoning system for locomotive diagnostics. *Engineering Applications of Artificial Intelligence.* **12** (6), pp. 681 - 690.

[Vila, 1994] Vila, L. (1994). A survey on temporal Reasoning in Artificial Intelligence. *AI Communications,* **7**, pp. 4-28.

[Virdell, 2003] Virdell, M. (2003). *Business processes and workflow in the Web services world.* [Online]. Available at: http://www.ibm.com/developerworks/webservices/library/ws-work/index.html. [Accessed 2 July 2010].

[W3C, 2001] W3C (2001). *Web Services Description Language (WSDL) 1.1.* [Online]. Available at: http://www.w3.org/TR/wsdl. [Accessed 18 March 2011].

[Wang et al., 2005] Wang, M., Wang, H., Xu, D. (2005). The design of intelligent workflow monitoring with agent technology. *Knowledge-Based Systems*, **18 (**6), pp. 257 – 266.

[Watson, 1997] Watson, I. (1997). *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. San Francisco, CA: Morgan Kaufmann.

[Weber et al., 2005] Weber, B., Rinderle, S., Wild, W., Reichert, M. (2005). CCBR-Driven Business Process Evolution. *6th International Conference on Case-Based Reasoning* (ICCBR'06). 23 - 26 Aug 2005. Chicago, IL, USA. Berlin: Springer – Verlag, pp. 610 – 624.

[WfMC, 2008] WfMC. (2008). *XPDL 2.1 Complete Specification.* [Online]. Available at: http://www.wfmc.org/xpdl.html. [Accessed 14 June 2010].

[Wolf & Petridis, 2008] Wolf, M., Petridis, M. (2008). Measuring Similarity of Software Designs using Graph Matching for CBR. In: *Workshop Proceedings of AISEW 2008* (ECAI 2008). Patras, Greece.

[Wooldridge, 2009] Wooldridge, M. (2009). *An Introduction to MultiAgent Systems* (2nd edition). England: Wiley.

[World Economic Forum, 2011] World Economic Forum. (2011). *The Global Information Technology Report 2010–2011 Transformations 2.0*. [Online]. Available at: http://www3.weforum.org/docs/WEF_GITR_Report_2011.pdf. [Accessed 2 February 2012].

[Wu, Harrigan & Cunningham, 2011] Wu, G., Harrigan, M., Cunningham, P. (2011). *A Characterization of Wikipedia Content Based on Motifs in the Edit Graph*. UCD-CSI-2011-02. School of Computer Science & Informatics, University College Dublin. Dublin, Ireland.