Workflow Monitoring and Diagnosis Using Case Based Reasoning on Incomplete Temporal Log Data

Stelios Kapetanakis, Miltos Petridis, Jixin Ma, Liz Bacon

School of Computing and Mathematical Sciences, University of Greenwich, Maritime Greenwich Campus, Old Royal Naval College, Park Row, Greenwich, London SE10 9LS, UK
(s.kapetanakis, m.petridis, j.ma, e.bacon}@gre.ac.uk

Abstract. This paper presents an approach for intelligent diagnosis and monitoring of workflows based on incomplete operation data in the form of temporal log data. The representation of workflows in this research using graphs is explained. The workflow process is orchestrated by a software system using BPEL technologies in a service oriented architecture. Episodic cases are represented in terms of events and their corresponding temporal relationships. The matching and CBR retrieval mechanisms used in this research are explained and the architecture of an integrated intelligent monitoring system is shown. The paper contains a simple evaluation of the approach based on a university quality assurance exam moderation system. Finally, further work on the system and the extension to an intelligent monitoring and process optimisation system is presented.

Keywords: Case Based Reasoning, Business Workflows, Temporal Reasoning, Uncertainty, Graph Similarity

1 Introduction

Modern business processes are increasingly being monitored and managed using computer systems. In order for this to happen effectively, business processes are more formally defined and structured events relating to their operation are captured and reported to the various business process stakeholders and managers.

Business processes are typically defined and represented in terms of a series of workflows and temporal relationships and constraints between them. Business processes can be defined using UML diagrams such as activity diagrams and represented formally using newly emerged business process representation standards. The Business Process Modelling Notation (BPMN) developed by the Business Process Management Initiative (BPMI) and Object Management Group (OMG) provides a standard for the graphical representation of workflow based business processes[1]. Workflow based business process representation is possible with standards covering the definition, orchestration and choreography of business processes.

Over the last few years, a number of standards have emerged and are widely accepted and supported by mainly Service Oriented Architecture (SOA) based enterprise technologies and systems. The OASIS Business Process Execution Language (BPEL), short for Web Services BPEL (WS-BPEL) is a key orchestration technology [2]. The Workflow Management Coalition (WfMC) backed XML Process Definition Language (XPDL) is a format standardised to interchange Business Process definitions between different workflow products and systems.

Modern enterprise systems are able to separate the definition of workflow based business processes from the software implementing the operation of these workflows, offering much more flexibility and agility than it was possible in older systems. This allows enterprise computer systems to monitor and control business processes and workflows within an organisation. Additionally, this allows for the agile change of workflows to adapt to changing business needs of the organisation.

Case Based Reasoning (CBR) has been proposed as a natural approach to the recall, reuse and adaptation of workflows and knowledge associated to their structure. Minor et al [4] proposed a CBR approach to the reuse and adaptation of agile workflows based on a graph representation of workflows and structural similarity measures. The definition of similarity measures for structured representations of cases in CBR has been proposed [5] and applied to many real life applications requiring reuse of domain knowledge associated with rich structure based cases [6],[7].

A key issue associated with the monitoring and control of workflows is that these are very often adapted and overridden to deal with unanticipated problems and changes in the operating environment. This is particularly the case in the aspects of workflows that directly interact with human roles. Most business process management systems have override options allowing managers to bypass or adapt workflows to deal with operational problems and priorities. Additionally, workflows are liable to change as the business requirements change and in many case workflows involving processes from different parts of an organisation, or between collaborating organisations can "tangle", requiring the need for synchronisation and mutual adaptation to allow for compatible synergy.

The flexibility and adaptability of workflows provides challenges in the effective monitoring of a business process. Typically, workflow management systems provide outputs in terms of event logs of actions occurring during the execution of a workflow. These could refer to an action (such as a sign-off action or uploading a document), or a communication (such as a transaction initiation or email being initiated and sent). The challenge in monitoring workflows using event information is that even where the workflow structure is well defined and understood, the trace of events/actions does not usually contain the context behind any decisions that caused these events/actions to occur. Additionally, there are often a lot of contextual information and communications that are not captured by the system. For example, some actions can be performed manually and informal communications/meetings between workflow workers may not be captured by the system. Knowledge of the workflow structure and orchestration of workflows does not necessarily define uniquely the choreography and operation of the workflows.

The effective monitoring of workflows is therefore required to deal with uncertainty stemming from these issues.

The approach proposed in this paper is based on a CBR process requiring similarity measures informed from knowledge discovery of norms and problems from past operation. The CBR approach proposed uses graph based representation of cases based on events, actions and intervals and their temporal relationships.

Section 2 discusses the exam moderation business process application domain that is used to evaluate the approach.

Section 3 presents the proposed workflow and event log case representation and similarity measures used.

Section 4 presents the architecture of the workflow intelligent monitoring system CBR-WIMS that has been developed to evaluate this work.

Section 5 presents an evaluation based on two workflow monitoring experiments.

2 The Exam Moderation Business Process Workflows

In order to evaluate the approach proposed in this research, it was decided to use the University of Greenwich, School of Computing and Mathematical Science exam moderation system. This is an automated web enabled secure system that allows course coordinators, course moderators, exam drafters (typically senior managers), admin staff and external examiners to upload, modify, approve and lock student exam papers. The system automates the whole process and provides an audit trail of events generated by workflow stakeholders and the system. The system orchestrates a formal process made up of workflows. The process can be defined and displayed formally in terms of a UML activity diagram (Fig. 1). The system tracks most workflow actions in terms of timed events. Most timed events, generate targeted email communications to workflow stakeholders, some for information and others requiring specific further actions from these stakeholders.

For example, the action of a new exam version upload from a course coordinator is notified to the moderator, drafter and admin staff. This can prompt the moderator to approve the uploaded version or upload a new version. However, the coordinator can also upload a new version and admin staff may also decide to format the uploaded version and upload it as a newer version. The system captures all versions, workflow actions, emails sent and there is a facility to record free form comments to document versions and/or workflow actions.

2.1 Uncertainty in Workflows

The overall exam moderation workflow process is formally defined and constrained by the system operation. There are also some limited facilities for manual override by the system administrator. However, the overall process in conjunction with the actions and communications audit trail do not uniquely explain the exact cause of individual actions and cannot predict reliably what the next event/action will be and when this is likely to occur. Most of the uncertainty stems from the problem

that a significant part of the workflows occur in isolation from the system. The system does not capture all of the contextual knowledge associated with workflows. A lot of the communications between workflow stakeholders can occur outside the system (direct emails, physical discussions and calls) adding to the uncertainty associated with past or anticipated events and the clear definition of the current state.

Discussions with workflow monitoring managers showed that patterns of events indicated, but not defined uniquely the current context and state of a workflow. Managers were able to guess from looking at the workflow events and communications audit what the context and current state of a workflow was and point to possible problems. Most problems occur due to human misunderstanding of the current state and confusion with roles and responsibilities and usually result to the stalling of a workflow. Managers will then try to restart the process by adding comments to the system, or initiate new actions and communications. However, this depends on managers realizing that such a problem has occurred.
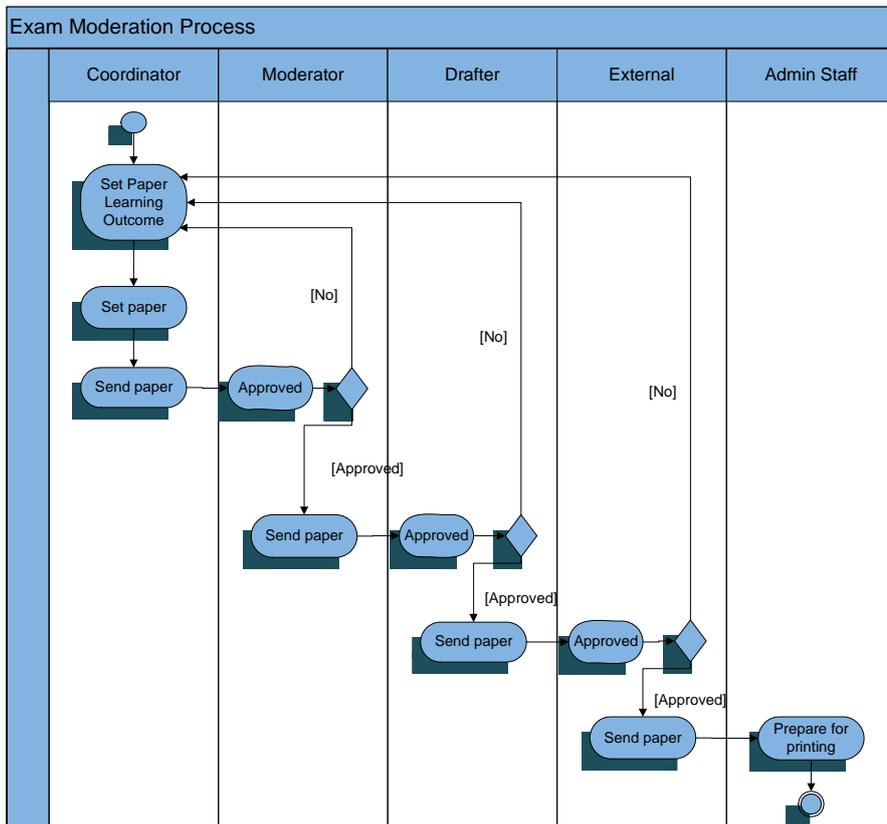
Fig. 1. The exam moderation process activities and workflows (simplified)

A typical problem series of event could be one where a stakeholder has missed reading an email requiring an action. In that case, the workflow would stall until a

manager or another stakeholder spots the problem and produces a manual action to get the workflow moving again. For example, a course coordinator upload notification may have been missed by a moderator who would then not read the new version and either approve or try to amend by a new upload as she needs to do. In that case, the coordinator may take no further action and other stakeholders will not act expecting an action from the moderator to occur.

A key problem with uncertainty about the current status of a workflow is that due to the expected normal delay between workflow events/actions, it may not be clear at a given point in time whether the workflow has stalled or the moderator is just slow at responding to the original action of the coordinator upload. This can only be resolved in a stochastic way based on retrieved knowledge from similar series of events in past workflows.

Discussions with system managers indicated that some of the uncertainty associated with expected response delays can be reduced by using past experience about response profiles and norms for individual stakeholders. Data mining or statistical analysis of the information obtained from past workflows for individual system users in a particular workflow role can provide the most likely response and likely response time for the user in a new workflow context. This can then be used to provide a more reliable similarity measure for the effective comparison between a new, unknown workflow state and past cases as part of a case-based reasoning retrieval process.

2.2   The CBR Workflow Monitoring System

The aim of the CBR Workflow Intelligent Monitoring System (CBR-WIMS) is to provide an automatic monitoring system that will notify managers and stakeholders of potential problems with the workflow and provide advice on actions that can remedy a perceived problem.

The monitoring system is designed to work based on experience of past event/action temporal sequences and the associated contextual knowledge and classification in a Case-Based Reasoning system. Similarity measures allow the retrieval of close matches and their associated workflow knowledge. This allows the classification of a sequence as a particular type of problem that needs to be reported to the monitoring system. Additionally, it is intended that any associated knowledge or plan of action can be retrieved, adapted and reused in terms of a recommendation for remedial action on the workflow.

The CBR monitoring system uses similarity measures based on a linear graph representation of temporal events in a workflow normalized by experience from past experience on individual user workflow participation patterns.

3 Workflow and Event Log Representation and Similarity Measures

In CBR-WIMS workflows are defined using UML activity diagrams and mapped through Business Process Management Notation (BPMN)[1] into Web-Services Business Process Execution Language (WS-BPEL) [2] and stored within the system. The storage of workflows is temporal as a number of versions can be stored to allow for modifications of the workflow following business process changes and its application to different contexts of use for a particular process. For example, variants of the exam process workflows can be defined to allow for specific types of exams, such as ones requiring external validation or collaboration for courses delivered collaboratively with other institutions. Similarity measures between workflow representations can be defined on a graph representation of workflow processes using an exhaustive graph similarity search algorithm based on the Maximum Common Subgraph [7]. This allows the reuse of knowledge about workflows between different workflow processes and variants. This is beyond the scope of the work presented in this paper.

The workflows stored in WS-BPEL are used by CBR-WIMS to automatically orchestrate the execution of workflows in the system.

The representation of events in the workflow event log is based on a general time theory based on intervals [8]. In the theory used here, the temporal relationships have been reduced from the ones proposed by Allen [9] to just one, the "meets" relationship.

The general time theory takes both points and intervals as primitive. It consists of a triad (T, **Meets, Dur**), where:

— T is a non-empty set of time elements;
— **Meets** is a binary order relation over T;
— **Dur** 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.
This approach has been shown to be suitable for defining temporal similarity measures in the context of a CBR system based on the graph representation of events and intervals and their temporal relationships and similarity measures based on graph matching techniques such as the Maximum Common Subgraph (MCSG)[11][7]. Additionally, such a graph can be checked for consistency of temporal references using linear programming techniques [11].
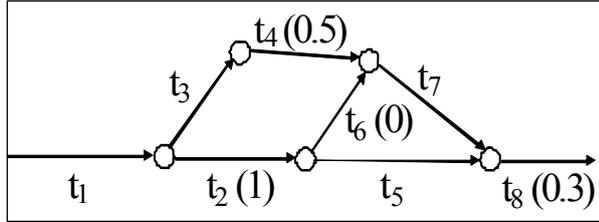
For example, consider a scenario with a temporal reference (T, M, D), where:

$T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9\}$;

$M = \{$Meets(t_1, t_2), Meets(t_1, t_3), Meets(t_2, t_5),
Meets(t_2, t_6), Meets(t_3, t_4), Meets(t_4, t_7),
Meets(t_5, t_8), Meets(t_6, t_7), Meets(t_7, t_8);

$$D = \{Dur(t_2) = 1, Dur(t_4) = 0.5,$$
$$Dur(t_6) = 0, Dur(t_8) = 0.3\}$$

The graphical representation of temporal reference (T, M, D) is shown in Fig. 2:

Fig. 2. Graph representation of temporal relationships

The Maximum Common Subgraph similarity between two such graphs can be defined as:

$$S(G,G') = \frac{(\sum_{\substack{matches \\ C,C' \\ in \\ MCG}} \sigma(C,C'))^2}{count(G).count(G')} \qquad (1)$$

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

In the case of time stamped events produced by the workflow event log, the duration of each interval can be calculated, so the graphs are collapsed 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 given order in each of the compared workflow logs. In this common graph segment each edge (event or interval) has a similarity measure to its counterpart in the other log that exceeds a given threshold value ε. Eq. 1 above can still be used to provide the overall similarity between the two workflows.

4   The Architecture of the Workflow Intelligent Monitoring System CBR-WIMS

CBR-WIMS is an Intelligent Workflow Monitoring System incorporating a CBR component. The role of the system is to assist the transparent management of workflows in a business process and to orchestrate, choreograph, operate, monitor and adapt the workflows to meet changing business processes and unanticipated operational problems and inconsistencies. Fig. 3 below shows the overall architecture

and components of CBR-WIMS. The system allows process managers to create, modify and adapt workflows to suit the changing business needs, and/or to allow for variations related to special business requirements. Workflow descriptions are stored in a temporal repository and can be used for looking up past business processes and to provide historical context for past event logs of operations.

The main part of the system controls the operation of the workflows. It responds to actions of various actors to the system and communicates messages about the operation of the system to them. The control system has an workflow orchestrator component that looks up the current workflow definition and orchestrates responses by invoking specific Web Services. The control component also manages and updates the data stored and current state of the workflow operation and provides event audit log of the key events and actions that occur within the operation of the workflow.

The workflow monitoring and intervention controller monitors, reports, and proposes possible remedial actions to the workflow operation manager. The monitoring system uses a CBR system to retrieve past useful experience about workflow problems occurred in the past by retrieving similar sequences of events/actions in the events log for a given workflow (or workflow part) compared to the current state and recent sequence of events/actions in the operation of the workflow. If a fault or possible problem pattern is detected, this is reported to the workflow operations manager together with the retrieved similar cases and associated recorded experience of any known remedy/course of action.
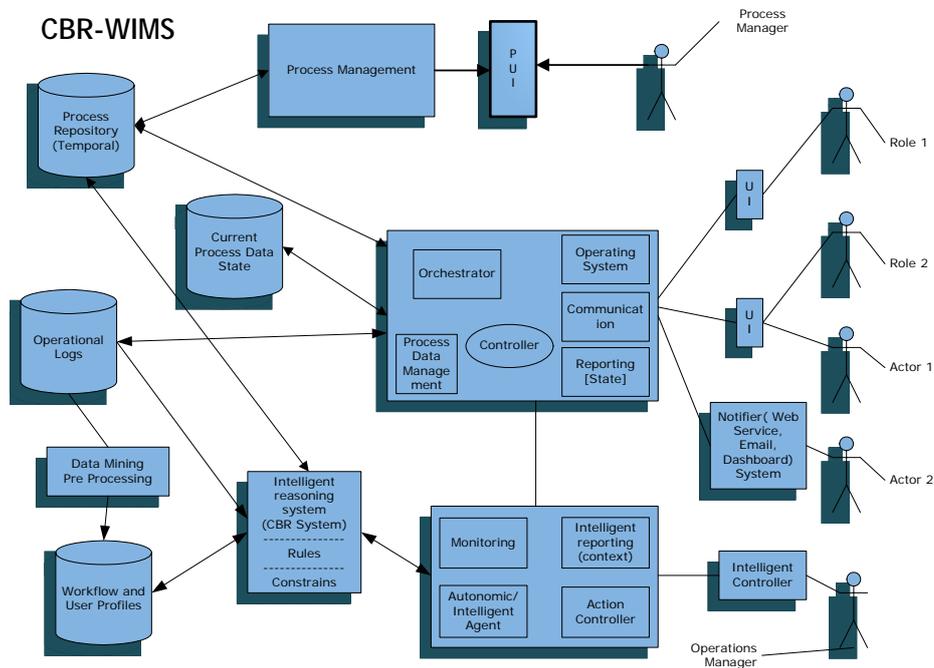
Fig. 3. The Intelligent Workflow Management System Architecture

In order to deal with the uncertain and contextual dimension of workflow similarity, the CBR system relies on knowledge discovered from past cases on workflow norms and user profiles created by statistical and data mining pre-processing. The pre-processing component analyses operational logs and attempts to discover knowledge about norms and patterns of operation that can be used in the calculation of the similarity measures for the CBR process. This is particularly important for the monitoring process as any "interesting" / "abnormal" states need to be seen in the context of what has been normal/abnormal behaviour in past event sequence cases.

5   Workflow Monitoring Experiments and Evaluation

In order to evaluate the suitability of the approach proposed in this paper, a number of simple experiments were conducted using the CBR-WIMS system. A simplified workflow process based on the exam moderation problem was constructed and a simulation was used to produce a series of workflow case studies. 320 simple event logs of workflows were produced to serve as cases in the case base. Each case was labelled as either "stalled" or "not stalled" to indicate the presence or not of a problem in the workflow execution. Only exam upload actions were considered and only the last 3 such uploads in a series of workflow events were used to represent each case.

A workflow event log audit trace is represented as:

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

An example of this would be (intervals are in days):

(CoordUpload,John,3, ModUpload, Phil, 0,CoordUpload, John, 5)

In the first instance the name of the person involved was ignored, focusing solely on the role involved in the action.

The similarity measure between two actions A_1 and A_2 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 two intervals I_1 and I_2 is defined as:

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

The Maximum Common Subgraph (MCSG) between cases C and C' is assembled starting right (latest) to left (earliest) calculating similarity measures matching each interval and action in C to the corresponding one in C', stopping when the similarity between two edges falls under a threshold set at 0.5.

For example, given 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 MCSG:

```
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 .. MCSG Matching stops
```

So, the overall similarity between C and C' from eq. 1 is:

$S(C,C') = (0.75+1+1+1+0.857)^2/6^2 = 0.59$

The 320 cases were split randomly into a case base of 300 cases and 20 test target cases. Using the KNN algorithm for K=3, the three nearest neighbours to every target case were used to classify the target case as "stalled" or "not stalled" using simple voting. The results were compared against the known classification for the target cases. This evaluation run was repeated 10 times and the results of the classification were averaged over the 10 runs.

Table 1. below shows the results of the evaluation runs:

	Average number of cases / 20	%
Target Cases Correctly classified	13.8	69
Missed positives	5	25
False positives	1.2	6

Table 1. First Evaluation results – no normalisation for person profiles

For the second set of experiments, the interval similarity measures were normalised to take into account the different rates of responses expected from different workflow actors. A Data analysis of the cases classified workflow actors into:

— Fast responders: 0-2 days
— Medium responders: 2-4 days
— Slow responders: over 4 days

For these cases, the interval duration I for each interval was replaced by the difference of the actual duration minus the nominal duration for the relevant type of workflow actor:

— Fast responders: 1 day
— Medium responders: 3 days
— Slow responders: over 5 days

So assuming that in the example above analysis of past behaviour has shown that John is a fast responder and Phil is a slow responder, the case is represented as:

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

This way the similarity measure is modified to provide a context based on knowledge discovered from past cases.

The results of running a similar set of experiments as in the first iteration are summarised in Table 2.

	Average number of cases / 20	%
Target Cases Correctly classified	15.3	76.5
Missed positives	3.8	19
False positives	0.9	4.5

Table 2. Second Evaluation results – normalised for person profiles

It can be seen that the overall number of target cases correctly classified has increased, mainly by corresponding reduction of missed positives.

This preliminary evaluation is encouraging. Further evaluation using a larger dataset from actual (not simulated) workflow event audit logs is planned to evaluate this approach further. In the planned work, larger segments of event log will be used in the case represenation involving the full set of possible exam moderation actions and events to predict the exact type of workflow disruption.

6   Conclusions

This paper discussed an approach for intelligent diagnosis and monitoring of workflows based on incomplete operation data in the form of temporal log data. This was based on a graph representation of workflows using temporal relationships. The workflow process is orchestrated by a software system using BPEL technologies in service oriented architecture in the CBR-WINS system. The matching and similarity measures presented here showed in a preliminary evaluation that they are capable to classify problems correctly in a simplified workflow process. In particular it was shown that an analysis of past workflow event logs can provide norms and context

that can reduce the uncertainty in similarity based matching and improve the efficiency of the reasoning process.

Further work will concentrate on further and more realistic evaluation of the approach based on more complex case representation and similarity matching. Work on further building and automating the CBR-WINS system will allow the extension to provide intelligent advice to operators in addition to the existing simple monitoring action. Other work direction will cover the challenge of explaining the reasoning results and advice to the workflow operation managers, the combination of constraints and temporal consistency checking and the combination of workflow event log temporal knowledge with other uncertain temporal knowledge available about a workflow.

Finally, the reuse of knowledge across different workflows, concentrating to changed workflows and variants can be investigated.

References

1. Business Process Management Initiative (BPMI): BPMN 1.1: OMG Specification, February, 2008, http://www.bpmn.org/,   accessed April 2009.
2. OASIS: BPEL, The Web Services Business Process Execution Language Version **2.0.** http://www.oasis-open.org/apps/org/workgroup/wsbpel, May 2006.
3. Workflow Management Coalition (WfMC): XPDL 2.1 Complete Specification (Updated Oct 10, 2008), http://www.wfmc.org/xpdl.html, accessed April 2009.
4. Minor, M., Tartakovski, A. and Bergmann, R.: Representation and Structure-Based Similarity Assessment for Agile Workflows, in Weber, R., O. and Richter, M., M.(Eds) Case-Based Reasoning Research and Development, Proceedings of the 7th international conference on Case-Based Reasoning, ICCBR 2007, Belfast, NI, UK, August 2007, LNAI 4626, pp 224-238, Springer-Verlag, 2007
5. Bunke, H., Messmer, B.T.: Similarity Measures for Structured Representations. In: Wess, S., Richter, M., Althoff, K.-D. (eds) Topics is Case-Based Reasoning. LNCS, vol. 837, pp 106-118, Springer, Heidelberg (1994)
6. Mileman, T., Knight, B., Petridis, M., Cowell, D., Ewer, J. (2002): Case-Based Retrieval of 3-D shapes for the design of metal castings in Journal of Intelligent Manufacturing, Kluwer. 13(1): 39-45; Feb 2002.
7. Wolf, M., Petridis, M.: Measuring Similarity of Software Designs using Graph Matching for CBR, In workshop proceedings of AISEW 2008 at ECAI 2008, Patras, Greece (2008)
8. Ma, J., Knight, B.: A General Temporal Theory, the Computer Journal, 37(2), 114-123 (1994).
9. Allen, J., Hayes, P.: Moments and Points in an Interval-based Temporal-based Logic, Computational Intelligence, 5, 225-238 (1989).
10. Ma, J., Knight, B.: A Framework for Historical Case-Based Reasoning, in K.D. Ashley and D.G. Bridge (Eds.): ICCBR 2003, LNAI 2689, pp. 246–260 (2003)
11. Ma, J., Knight, B., Petridis, M.: Deriving Explanations from Partial Temporal Information, In workshop proceedings of ExACT-08 at ECAI 2008, Patras, Greece (2008)