

2011

The Scenario-Oriented Method for Recording and Playing-Back Healthcare Information

Yi Ding

Technological University Dublin, d08117157@mydit.ie

Bing Wu

Technological University Dublin, bing.wu@tudublin.ie

Erqiang Zhou

Technological University Dublin, erqiang.zhou@tudublin.ie

See next page for additional authors

Follow this and additional works at: <https://arrow.tudublin.ie/scschcomcon>



Part of the [Health Communication Commons](#)

Recommended Citation

Ding, Y., Wu, B. & Erqiang, Z. (2011). The Scenario-Oriented Method for Recording and Playing-Back Healthcare Information. *Proceedings of CENTERIS*, (M.M. Cruz-Cunha et al. (Eds.)): Part III, CCIS 221, pp. 175–184, 2011. doi:10.1007/978-3-642-24352-3_19

This Article is brought to you for free and open access by the School of Computer Science at ARROW@TU Dublin. It has been accepted for inclusion in Conference papers by an authorized administrator of ARROW@TU Dublin. For more information, please contact arrow.admin@tudublin.ie, aisling.coyne@tudublin.ie, vera.kilshaw@tudublin.ie.

Authors

Yi Ding, Bing Wu, Erqiang Zhou, and Jianfeng Wu

The Scenario-Oriented Method for Recording and Playing-Back Healthcare Information

Yi Ding, Bing Wu, Erqiang Zhou, and Jianfeng Wu

K-Camp, School of Computing, DIT, Kevin Street, Dublin 8, Ireland
{yi.ding,jianfeng.wu}@mydit.ie, {bing.wu,erqiangzhou}@dit.ie

Abstract. This paper proposes a new method, called the *scenario-oriented method*, to support the idea of recording and replaying the healthcare information such that the reporting and decision-support capabilities can be enhanced. In order to play back the changing history of certain information units, the scenario-oriented method attempts to organize related information and knowledge elements as a *context* so that the history of real medical activity can be recorded, and then be queried as a continuous, on-the-fly, understandable and playing-back information scenario through replay operations.

Keywords: Scenario-oriented, Replaying, Healthcare.

1 Introduction

Due to the increasing demand for reduced cost and improved quality of service in healthcare, the computerization of *Clinical Practice Guidelines* (CPGs) has been promoted. For computerized information systems implemented to manage such CPGs, a key goal is to facilitate the process of retrieving specific information unit so as to review its whole changing history, such as, replaying a medical patient plan produced during the treatment of disease according to certain clinical guidelines. The *replaying* ability enhances the reporting and decision-support capabilities in the organization. Current large-scale information systems are designed to support the general queries (e.g. querying relational data, or retrieving parts/whole of documents, etc.) and lack of the ability to review the history information. As a result, users are often unable to obtain the history of a desired specific information unit within a well-defined subject area. This fact led to the proposed approach in this paper so as to enhance the specific querying capabilities (replay) of computerized information system.

This paper proposes a new method, called *scenario-oriented method*, which attempts to organize/reorganize all medical information and knowledge elements that would constitute a meaningful domain information scenario such that the real medical activities can be recorded and then replayed dynamically in future. One of the benefits of this method is that users can effectively view the day-to-day medical activities as a continuous, on-the-fly information scenario through a *Replay Scenario* statement, which combines a query-response approach and information visualization techniques.

The rest of this paper is organized as follows: Section 2 presents the related work. Section 3 gives an overview of the scenario-oriented method. Section 4 gives a Microalbuminuria Protocol (MAP) case study. Section 5 outlines an implementation architecture to realizing the scenario-oriented method. Section 6 concludes this paper.

2 Related Work

Since 1980s, research in Human Computer Interaction (HCI) has used scenarios as representation of system requirements to improve the communication between developers and users [1]. There is a great variety of scenario usages in many different disciplines. One typical usage is scenario-based information retrieval, which has been extensively explored in the field of seeking information from medical text [2,3,4,5]. In retrieving medical text, users are often interested in answers relevant to certain scenarios, scenarios that correspond to common tasks in medical practice. Current information systems do not provide efficient way to fulfil this requirement.

Facing with this challenge, Becher et al. [2] presented a user interface of summarization system for physicians in Bone Marrow Transplantation. The interface has users' state well-articulated questions that serve as summarization target and basis for retrieval queries, and it displays summarization results in an organization that fits the user's situation. A scenario forms with intermediary structures, which is a detailed interpretation for summarizing and an abridged one for IR, has been developed to mediate between the users' and the system perspective. Within these interpretations, the form itself is represented by constants, the user query provides variables.

Chu et al. [3] developed a scenario-based digital medical library which consists of scenario-based proxies, context-sensitive navigation and matching, content correlation of documents, and user models. The use of scenario based proxies here refers to relevant information can be obtained by direct access and similar-to search or content correlation links of the information sources. By using scenarios, this research aims at providing more focused searching thus to be more specific to the patient's medical condition.

Liu et al. [4,5] proposed a knowledge-based query expansion method that exploits the UMLS knowledge source to append the original query with additional terms that are specifically relevant to the query's scenario. Here, scenario-specific queries represent a special type of queries that frequently appear in medical free-text retrieval. That is, a scenario consists of several existing queries. This paper only used a knowledge-based query expansion method to improve the retrieval performance for such kind of queries.

Except for using scenario in retrieving medical text, Strasunskas and Tomassen [6] proposed a scenario-driven information retrieval approach to complement rule-based monitor of subsea production, which could be seen as a kind of view-based search. Because not all possible cases can be encoded in rules beforehand and precision of retrieving information is very important, the information retrieval should be adjusted to the scenario. Strasunskas and Tomassen first described the information and knowledge resources that enable users to formulate task-specific queries and then presented how system automatically formulates a query that is sent to a vector-space model information retrieval engine.

In Shen et al. [7], a scenario-oriented recommendation system was developed to help people to determine the ideal productions even when users don't necessarily know exactly what product characteristics they are looking for. Shen et al. used a common-sense reasoning system to map between the goals stated by the user, and possible characteristics of the product that might be relevant. Scenario-oriented recommendation breaks down boundaries between products' categories, finds the first example for existing techniques, and helps promote the independent brands.

Overall, a very few reviewed search methods provide the ability to review the changing history of specific information units. More importantly, the reviewed approaches don't provide the support for end users to dynamically construct the complete information scenario formed by the effects of the data/information changes and to record & replay the information scenario as a whole. Experienced from our earlier work [8,9,10,11,12,13,14] suggests that presentation, analysis and review of information can be significantly simplified and enhanced through a new paradigm that combines information visualization with allowing information to be managed, played and replayed in a dynamic and interactive manner on the basis of a formally constructed and meaningful domain information scenario.

3 An Overview of Scenario-Oriented Method

In this research, a scenario is defined as *a structure description of multi-dimensional information clusters to express the sequence and/or linkage of real practices computerised in an information system.*

The basic idea behind the *scenario* is that, in order to trace the changing history of certain information unit, the specific information unit should be put into *context* to reveal how it evolved and how it interacted with other information units in an application domain. Therefore, the default semantics of scenario, which is used to represent the real medical activity, is interpreted as “*who did what operated by whom in what time and where, under what condition and for what reason*”. Scenario can be either *simple* or *complex*. A simple scenario just describes one specific medical activity, such as, *patient_take_urine_test*. But it can't give global descriptions of an application domain. Thus, a complex scenario is developed to represent a complex activity of a certain application domain. For example, a scenario about a patient therapy history is a complex scenario, which consists of several test-taking events or other treatment events. A complex scenario is composed of different existing scenarios (simple or complex) in order to show the dependencies and interactions in the application domain.

In order to facilitate the process of replaying healthcare information by following the scenario-oriented method, a scenario model together with a specification language is developed. This scenario model is mainly used to specify what to be recorded and how they are linked – this will be helpful for the future play-back and query as the structural link at the conceptual level of the information unit are specified (so a querying path at conceptual level can be decided at this level without concerning to logical or lower levels).

A scenario language is also investigated and developed to support the scenario-oriented method. The scenario language consists of two main components: a specification component and a query component. The specification component provides the ability to specify scenarios by following the structure of scenario. The query component provides a scenario-oriented query language that is used to manipulate and query scenario; it also supports the concept of replaying information scenario, by using a replay scenario statement in high-level and combining information visualization techniques, to help users to understand the domain information. The scenario language is a high-level, declarative and XML-based language. The grammar syntax is defined using the XML schema, and the scenario specifications are represented as an XML document.

4 Applying the Scenario-Oriented Method to Healthcare Information: A Case Study

4.1 Description of the Microalbuminuria Protocol

The MAP, as shown in Figure 1, is a clinical protocol for the diagnosis and treatment of microalbuminuria in diabetes patient [9].

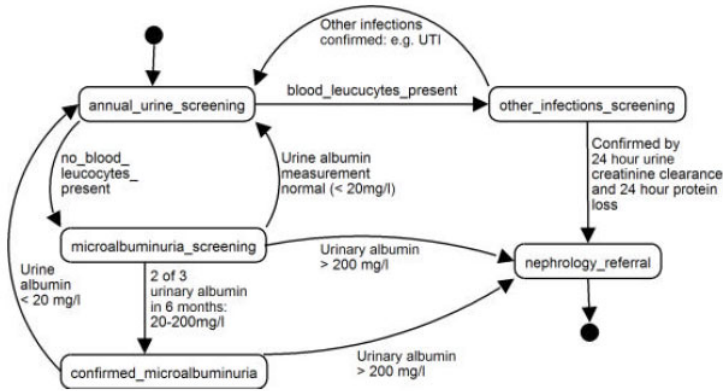


Fig. 1. State Chart for the Microalbuminuria Protocol

In the MAP case, there are several medical activities executed during disease management. These activities can be classified into: when patients are in the state of `annual_urine_screening`, they take Dip-Stick Urine (DSU) test; when patients are in the state of `other_infections_screening`, they take Urine Track Infection (UTI); patients take 24 Hour Creatine Clearance and Protein Loss (24CRCL_PL) test based on the result of UTI; when patients are in the state of `microalbuminuria_screening`, they take Albumin Creatine Ratio (ACR) test and Serum Creatine Ratio (SCR) test both; when patients are in the state of `confirmed_microalbuminuria`, three activities Optimizing Glycaemic Control, checking Blood Pressure (BP) and prescribing Angiotensin Converting Enzyme Inhibitors (ACE) will be executed for patients; when patients are in the state of `Nephrology Referral (NPH)`, they will get a Referral Note. The medical activity “patient take urine test” will be employed as a main example to explicitly illustrate how to use the scenario-oriented method for recording & playing back healthcare information. The more details of MAP can be found in [9] and [11].

4.2 Applying the Scenario-Oriented Method to Healthcare Information

Figure 2 shows the overall strategy of the scenario-oriented method for recording & playing-back healthcare information. Our method starts with requirement specification by using the structural model to express the components, attributes, and inter-relationships within the system. The structural model is a ‘network model’, which could be seen as layered and connected ER model.

Query Vocabulary. The *query* vocabulary will classify the identifiers (entities/objects/relationships) referred in the information system into six categories so as to reflect different angles/aspects of information scenarios. Such as, for a treatment activity that patients take urine test, the entity *patient* and entity *doctor* will be classified into *WhoList* to represent the people who execute the activity. The relationships *take* and *executed_by* will be classified into *ActionList* to represent the action operated by people. *WhatList* includes entity *urine test*. Attribute *examine.begin_time* and *examine.end_time* will be classified into *WhenList*. Attribute *examine.location* is in *WhereList*. The constraint *dsu_constraint* (only the status of patient is annual_urine_screening, patient could take dip-stick urine test) and rule *r_dsu* (for the end of year, patient takes the urine test) are both classified into *RuleList*. Furthermore, each identifier classified into query vocabularies will be specified in the form of database schema so as to provide a way to store the information about these identifiers into data repositories.

Scenario Space. The *scenario space* defines/includes Pre-defined *scenarios* of concern, of which certain aspects will be recorded, and can be played-back during the running-time of the information system. It is constructed based on the structural model and query vocabulary by following the formal structure of scenario. The scenario space for the *patient_take_urine_test* is presents in Figure 3. This is a personalized diagram derived from the structural model for MAP, which is only respect to the real medical activity about “patients take the urine test” in healthcare domain. And also, the elements *operated_by*, *whom*, *constraint* and *cause* are added into this scenario space to capture the full semantics of a scenario from different angles. Thus, the semantics represented by this scenario can be interpreted as “*patient take urine test executed by doctor, in examine.location, from examine.begin_time to examine.end_time, in the condition of the dsu_constraint and caused by r_dsu*”. The tag labelled for each block in scenario space, such as *who* in *patient* block, has two functionalities. One is to reveal this identifier selected from which vocabularies. The other semantics of tag is that, when labelling the tag at the upper left corner of block, the history information about this identifier will be recorded for later playing-back.

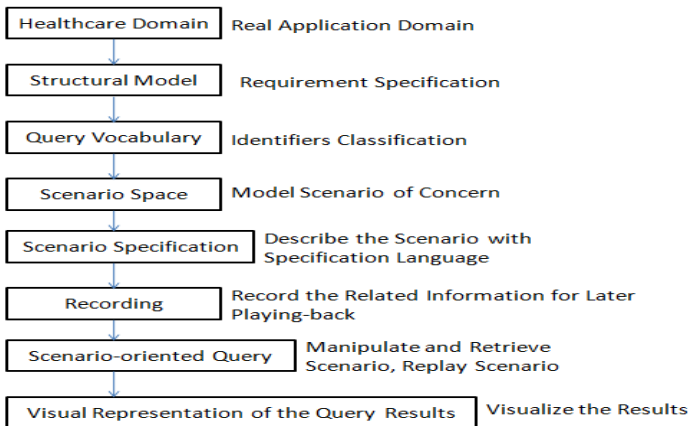


Fig. 2. The Flow Chat for Scenario-oriented Method

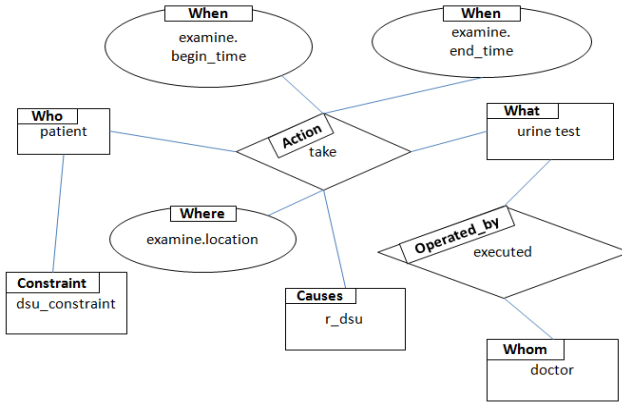


Fig. 3. Scenario Space for Patient Take Urine Test

Scenario Specification Language. In this research, scenario can be represented either by a graphical modelling (scenario space) or by a *scenario specification* language. Figure 4 shows the description of scenario “patient take urine test” that users are interested in. The *Header* element provides the descriptive information regarding a simple scenario. The *Id* element is used to provide unique identification for this scenario. The *Author* element presents the people who designed this simple scenario. The *Access_Right* element represents the authority of this scenario. Each simple scenario has a life-cycle in the system. Therefore, the *Design_Time* is used to describe the beginning time of simple scenario and *Terminate_Time* represents the end time of this scenario. The *Cause* element is used to specify the rule that starts the execution of process. It interprets “for what reasons” in the semantics of scenario. The *Constraint* element specifies some rules to standardize a process. It interprets “under what conditions” in the semantics of scenario. The *Process* element is the main component of simple scenario. It interprets “who did what by whom, in what time and where” in healthcare domain. Since the syntax of scenario specification language is defined by XML Schema, the description of scenario could be expressive to specify the user-centred domain knowledge not by programming language, and these information clusters could be managed by computerized approach.

Recording. While specifying the scenario *patient_take_urine_test* and labelling the tag for its identifiers, the related information will be specified to be recorded. The recording mechanism divides these information clusters into two parts and employs different methods to store them separately. The *domain information recording* will record the information sources of identifiers referred in this scenario. When an identifier is tagged in the scenario space, such as tag *whom* for *doctor*, the system begins to record information about this identifier. Each identifier (e.g. doctor) will have many instances (e.g. Jan, Patrick) with real data values in data repository. And for each instance, its whole changing history will also be recorded by default. The *scenario recording* is used to record the relationships among identifiers that constitute the scenario *patient_take_urine_test*. This scenario specification (shown in Figure 4) will generate many instances (e.g. Peter take his urine test) obtained by a XML document.


```

<SScenario name="patient take urine test">
  <Header>
    <ID>as00001</ID>
    <Designer>Yi Ding</Designer>
    <Design_time>2010-11-3</Design_time>
    <Terminate_time>Now</Terminate_time>
    <Access_Right>Write</Access_Right>
  </Header>
  <Causes>
    <Cause ref="r_dsu">
  </Causes>
  <Constraints>
    <Constraint ref="dsu_constraint">
  </Constraints>
  <Process>
    <Who ref="Patient">
    <Action ref="take">
    <What ref="urine test">
    <Operated_by ref="executed">
    <Whom ref="doctor">
    <When>
      <From ref="examine.begin_time">
      <To ref="examine.end_time">
    </When>
    <Where ref="examine.location">
  </Process>
</SScenario>

```

Fig. 4. The Specification for Scenario *Patient_Take_Urine_Test*

The instance of identifier referred in the scenario instance is specified by its ID in the XML document. When retrieving this scenario, the system will access real values of the object instance pointed by its ID from database directly and represent them under the structure specified by the scenario.

Scenario-oriented Query. After recording the needed information, how to formulate queries in the terms of path through scenario space to trace the changing history of specific information unit is the next phase of our method. The comprehensive query clause **REPLAY SCENARIO** statement is adopted here as the main tool to formulate the specific history queries. Based on the scenario space shown in Figure 3, several history queries could be expressed from different angles in high levels. These questions include: “what...?”; “where...?”; “when...?”; “who...?”; “how...?”; “why...?” or the whole evolution process. Some examples are presented as following:

Task: who did take the urine test, in what time, where and executed by whom?

Command: **REPLAY Scenario**[name="patient_take_urine_test"]
FROM Who, When, Where, Whom;

Task: When did Peter take what test, the reason and what’s the test result?

Command: **REPLAY Scenario**[name = "patient_take_urine_test"]
FROM What, Cause
WHERE Who[name = "Peter"];

In MAP case, one of the most complex question is that “how to review a patient’s therapy history of MAP”. Assume that we have specified the needed simple scenario, such as patient take urine test, moreover, the complex scenario represents the therapy history is also constructed. Thus, the command for querying this question can be seen as follows:

Task: Review the Peter’s therapy history of MAP;

Command: **REPLAY Scenario**[name="patient’s_therapy_history_of_MAP"]
WHERE Who[name="Peter"];

Some expected results are presented as follows:

Peter takes DSU (result is negative) executed by Jan, in Lab 201 from 2010-12-1 10:00 to 2010-12-1 10:34, for the reason: the end of 2010, Peter take DSU test (rule *r_dsu*);

Peter takes ACR (result is 60mg/l) executed by Jan, in Lab 202 at 2010-12-2 11:00, under conditions: Peter.state = micro-albuminuria-screening, for the reason: if the result of DSU is negative, take ACR test;

Peter get referral note (nephrology) created by Kevin, in Lab 101 at 2011-6-4 12:05, under conditions: Peter.state = NPH, for the reason: CMA6 (if ACR>200mg/l, place patient into NPH state) and NPH1 (if the state of patient is NPH, then create referral note for patient).

Combining with query-response approach and information visualisation technology, and also considering the content comprehension to human sense, we will list our results as writing a set of sentences in the form of “who did what by whom, in where and when, under what conditions and for what reasons”.

5 An Architecture to Implement the Scenario-Oriented Method

Figure 5 shows a proposed implementation architecture, which demonstrates the applicability of the method, to support the idea of playing-back healthcare information within the context of scenario. There are six main components in the architecture: 1) an user interface; 2) a scenario modelling component; 3) a recording component; 4) a real-time information system; 5) a scenario language component; 6) a replaying component.

The *real-time information system* component is mainly used to execute the real medical activities in a computerized way. The *User Interface* component provides the functionalities for end users to manage the real-time information system. It also enables users to construct and update the scenario model representing their specific history queries on healthcare information. In addition, the *User Interface* provides the functionalities to manage the life-cycle of scenario as well as replay the changing history of focused information units and visualize the results to make it human understandable (an example of visualizing the replaying history information can be found in [14]). The *Scenario Modelling* provides a recording mechanism to specify what kind of information needed to be recorded at conceptual level. In addition to this, it also enables end users to construct their specific *scenario* by integrating the different information and knowledge elements together based on certain logical relationships. The *Recording* component provides the ability to implement the recording mechanism to store the information clusters specified to be recorded into data repository by using DBMS instance. Domain information recording and scenario recording are employed to record the needed information from two directions. In order to manage the scenario in a computerized way, the *Scenario Language* component is mainly used to support the scenario specification, manipulation, and querying. Furthermore, comparing to the static text information provided by using the SELECT statement, the *Replay* component has the ability to translate the concerned scenario streams into a visual representation so as to take advantage of the human eye’s broad bandwidth pathway into the mind to allow users to see, explore, and understand the healthcare information.

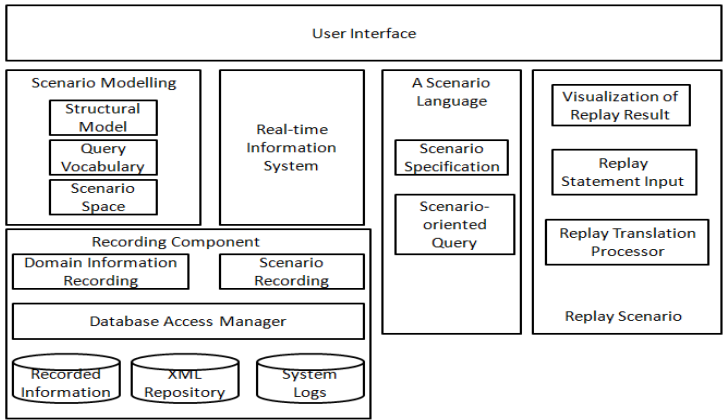


Fig. 5. An Implementation Architecture

6 Conclusion

Computerizing best practice in the form of healthcare information not only involves the domain knowledge specification and execution, but also involves the ability of replaying to enhance the reporting and decision-support capabilities in the organization. This paper proposes a *scenario-oriented method*, for supporting the idea of playing-back healthcare information. The healthcare domain is of particular benefit to the employing of scenario-oriented method. As this method is for supporting the playing-back of general *complex information*, it would also be of help to other application domains, such as aviation, personal information management, etc.

This work presented here is part of on-going research. As to the next step of this research, a prototype system with more comprehensive case study will be implemented to demonstrate the applicability of the scenario-oriented method and the implementation architecture.

References

1. Jarke, M., Bui, X., Carroll, J.: Scenario Management: An Interdisciplinary Approach. *Requirements Engineering* 3(3-4), 155–173 (1998)
2. Becher, M., Endres-Niggemeyer, B., Fichtner, G.: Scenario forms for web information seeking and summarizing in bone. In: *COLING 2002: Proceedings of the 2002 Conference on Multilingual Summarization*, pp. 1–8 (2002)
3. Chu, W., Johnson, D., Zhu, Q., Kangerloo, H.: Scenario-based information retrieval in a medical digital library. In: *International Conference on Digital Libraries: Research and Practice*, Kyoto, pp. 440–448 (2000)
4. Liu, Z., Chu, W.W.: Knowledge-based query expansion to support scenario-specific retrieval of medical free text. *Information Retrieval* 10(2), 173–202 (2007)
5. Liu, Z., Chu, W.W.: Knowledge-based query expansion to support scenario-specific retrieval of medical free text. In: *SAC 2005: Proceedings of the 2005 ACM Symposium on Applied Computing*, pp. 1076–1083 (2005)

6. Strasunskas, D., Tomassen, S.L.: Scenario-Driven Information Retrieval: Supporting Rule-based Monitoring of Subsea Operations. *Information Technology and Control* 36(1A), 87–92 (2007)
7. Shen, E., Lieberman, H., Lam, F.: What am I gonna wear?: scenario-oriented recommendation. In: *IUI 2007: Proceedings of the 12th International Conference on Intelligent User Interfaces*, pp. 365–368 (2007)
8. Wu, B., Kudakwashe, D.: PLAN: a framework and specification language with an event-condition-action (ECA) mechanism for clinical test request protocols. In: *Proceedings of the 34th Annual Hawaii International Conference on* (2001)
9. Kudakwashe, D.: A Generic Approach to Supporting the Management of Computerised Clinical Guidelines and Protocols. PhD Thesis. Dublin Institute of Technology (DIT), Dublin (2004)
10. Wu, B., Mansour, E., Dube, K.: Complex Information Management Using a Framework Supported by ECA Rules in XML. In: Paschke, A., Biletskiy, Y. (eds.) *RuleML 2007*. LNCS, vol. 4824, pp. 224–231. Springer, Heidelberg (2007)
11. Mansour, E.: A Generic Approach and Framework for Managing Complex Information. PhD Thesis. Dublin Institute of Technology (DIT), Dublin (2008)
12. Wu, B., Kudakwashe, D.: The Motion Picture Paradigm for Managing Information - A Framework and Approach to Supporting the Play and Replay of Information in Computerised Information Systems. In: *ICEIS*, pp. 515–518 (2008)
13. Mansour, E., Höpfner, H.: Replay the Execution History of Rule-Based Information. In: *Proceedings of the 2009 First International Conference on Advances in Databases, Knowledge, and Data Applications*, pp. 28–35 (2009)
14. Zhou, E., Wu, B., Ding, Y., Wu, J.: Coins: A modelling approach for recording & replaying complex information. In: *Proceedings of The IADIS International Conference Information Systems 2011*, pp. 37–44 (2011)