Journal of Scientific & Industrial Research
Vol. 69, March 2010, pp. 188-193

Exception detection in business process management systems

Dragan Mišic[1*], Miloš Stojkovic[1], Dragan Domazet[2], Miroslav Trajanovic[1], Miodrag Manic[1] and Milan Trifunovic[1]

[1]Faculty of Mechanical Engineering, Niš, A. Medvedeva 14, 18000 Niš, Serbia

[2]Faculty of Information Technology, Tadeuša Košæuška 63, 11000 Beograd, Serbia

Received 03 July 2009; revised 05 January 2010; accepted 08 January 2010

This study presents exception detection in process management system, which uses active semantic model (ASM), a hybrid system comprising semantic networks, expert systems, object oriented programming and fuzzy logic. ASM is employed to capture semantic features of all kinds of workflow situations, as well as to detect and interpret exceptions.

Keywords: Active semantic model (ASM), Exception, Workflow

Introduction

Workflows are managed by a workflow management system that defines, creates and manages workflows with software working within one or more workflow engines, capable of interpreting process definition, communicating with participants, and when necessary calling other IT tools and applications[1]. An exception[2] (expected and unexpected) during workflow is a situation that prevents execution of an activity or whole workflow in a defined way. Another classification of exceptions is done according to possibility of detection[3]. Casati[4] used event-condition-action rules for modeling exceptions. Active Semantic Model[5] (ASM) can be used for detection of exceptions. Expected exceptions are easily built into system using graphical users' interface. Unexpected exceptions are detected using various conclusion algorithms embedded in the system.

This study presents use of ASM for automatic detection of unpredicted exceptions in workflow of liquid-pump production process.

Active Semantic Model (ASM)

ASM comprises of semantic networks, expert systems, object oriented programming, fuzzy logic, group, set and graph theories (especially morphisms), etc. It is used to capture semantic features of all kinds of workflow situations, as well as to detect and interpret exceptions leading to workflow disturbances. It consists of three elements[5,6] [concept (CPT), association between CPTs, and body (record) of CPT]. CPTs are nodes of semantic network and interconnected with association, which is main element of knowledge base data structure. Association includes information about semantic relation of two CPTs in certain context, and indicates track of semantic interpretation of data. Body of CPT can include simple data (alpha-numeric symbols, variables and their values, series, scripts etc.), as well as complex structures of data (tables, object structures and execution codes). Contexts in ASM are used to mark group of semantically close associations. Situation, a context unchangeable in time, is a combination of associations between CPTs and other contexts independent of time. Event, a context changeable in time, is a combination of associations that change over time. Happenings are time sequence of a group of more or less similar situations.

Similarities of Concepts

Process of semantic categorization and interpretation of data is based on determining class and level of similarity between associations[7]. In determining similarity between associations, a motive describes why semantic interpretation is done. A motive in ASM is also a CPT in a context, connected to other CPTs. If two CPTs are similar, it is probable that other associations, owned by known CPT and not by a new one, can be applied to new CPT as well. For example, two CPTs (CPT_N & CPT_M) are not directly connected, but both

*Author for correspondence
E-mail: draganm@masfak.ni.ac.rs

connected through CPT_K. Higher degrees of similarity between two CPTs ($CPT_N \sim CPT_M$) depend on equality or similarity of other attributes of association to CPT_K.

Similarities of Contexts

Result of process of establishing similarities of contexts is semantic categorization of a new context compared to other known ones. Consequence of context categorization brings ASM to suggest creation of additional associations in a new context, reflecting contexts of same category. Semantic categorization of contexts by ASM enables directed and accelerated search of data in semantic interpretation. Contexts gather semantically close CPTs and their semantic relationships and thus divide data space. Associations (semantic relations) between CPTs represent signposts in searching Another result of determining similarity of contexts is eventual recognition of topological analogy (degree of homeomorphism of graphs of associations) between contexts not directly connected. Topological parameters of associations are type, roles, character and direction of association. Beside topological parameters, difficulty parameters of an association (significance and accuracy) are also considered. ASM determines whether contexts are possibly topologically analogous by this procedure.

Use of ASM in Exception Detection

Main module of process management system is core that manages workflow and expert system, which tries to solve situations that appear from exceptions in workflows. Open-source system, Shark, was used as system core[8], which takes process definition in XPDL format as an input. After that, it is possible to start several process instances, based on same definition. If an exception occurs during process execution, information about it is sent to expert system that tries to solve it. Solutions offered by expert system can bring a change in process definition or process instance.

Categorization of Exceptions by Way of Detection

According to possibility of detection in MD system, exceptions can be categorized as exceptions related to: i) data; ii) time (violation of dead-lines); iii) resources; and iv) signalized by people. MD system user can detect exception using special control parameter, which is defined for each process and can have one out of two values (continue and stop). Default value of this parameter is ‘continue’ and it stays until process moves on properly. If at some point, user establishes exception, value of this parameter is changed to ‘stop’, which is sign to start procedure of solving exception.

Detection of Exceptions Related to Data

Values and presence of different data and information can in certain situations represent exception. MD system uses ASM for detection of these exceptions. Categorization from standpoint of direct data processing represents formation of an association between context, which describes new situation, and “exception”, which already exists in ASM network.

Determining Context Similarity by Semantic Contents

Topologically correspondent associations (TCAs) belong to different contexts having same topologically parameters and difficulty parameters. Topologically correspondent concepts (TCCs) belong to TCAs, and have same role in those associations. During categorization, ASM between topologically analogous contexts (TACs) creates an association with following characteristics: role 1 = similar context; role 2 = similar context; type = topological_analogy; character = +; direction = both ways; significance = Savg; and accuracy = Havg. TACs are contexts of same type of topology or morphisms. In last step, ASM determines degree of topological similarity (number of TCAs and terms) and semantic closeness (number of mutual nodes) of association. Final result of determination of context similarities according to their contents is extraction of TACs to introduced context.

Context Upgrade Procedure

Every combination can be observed as a part of a larger semantic network, which is connected to other nodes in that network. Users introduce ASM to new association combinations by establishing associations between that combination or its nodes and other nodes in network, out of which some are already known to ASM and some are new. Each conclusion that ASM reaches is reflected in formation of new associations between CPTs of new combination and “known” network CPTs. Upgrade procedure of a new combination is based on similarity of new and known combinations. Context upgrade is conducted in two steps. Associations that a context as a whole has with other contexts and terms in the network are examined in first step. If a similar context has some associations that new one does not have, then establishing of associations between new context and known concept is proposed. Second step comprises combinations upgrade after example of a TAC.

For determination of association parameters, introduced network node is CPT_X (that is a context in following example). If nodes CPT_K and CPT_N exist so that CPT_X with CPT_K has an association of similarity class higher than 4 and if CPT_K and CPT_N nodes are connected by an association, then ASM proposes establishing of an association between CPT_X and CPT_N. Type and roles of new association are same as type of association between CPT_K and CPT_N, while other parameters are determined according to patterns. ASM conclusion in this case is establishing of an association between CPT_X and CPT_N, to which its similar term CPT_K is connected.

Example of Valve Manufacturing

In MD system, graphic process editor JaWE (Java Workflow Editor) is used for creating models. Process description is in XPDL format, which is then transferred to MD workflow management system. System administrator initiates new process instance based on that definition. After process instance initiation, all data defined for the process is created based on process definition. This data is input into common memory, which has been set aside for that process instance. If a new instance is being initiated for first time based on that definition, then it is necessary to define associations upon which ASM could later draw some conclusions. Association definition is done by administrator before execution of first activity of the process. All data from process definition is shown to administrator. This data represents CPTs in ASM terminology. It is upon administrator to connect proposed CPTs. When connecting, administrator also has possibility to define new CPTs, which are needed to describe actual process.

In valve manufacturing, body is made by casting and afterwards by machine processing. In casting process, one can get parts of an improper structure. In such situations, it is possible to try fixing parts. However, if that is not possible, those cast parts are thrown away. Detection of casting irregularity cannot often be done in advance, but those irregularities are realized in processing phase, when one start taking off layers of material. Appearance of porosity in casting signalizes exception. In that case, corrective actions should be undertaken. From standpoint of the process, actions can relate to adding new activities (welding), removing or changing existing ones. An expert system takes care of process changes, and here focus is on how a system can identify exception. Observed porosity is input into the system as new data. Simultaneously, position of that data, as a CPT, should be defined within ASM. After input of new data, defined ASM model is automatically presented to user (normal situation). Then user defines associations of new data with existing CPTs. Apart from defining associations of new with existing CPTs, user also defines that it is new CPT in question ('exception in valve manufacturing'). Since new context has some added CPTs in relation to normal context, it thus represents subtype of normal context. ASM recognizes this automatically and establishes an adequate type-subtype association between these contexts (Fig. 1). Appearance of porosity represents an exception in valve machining. New context is therefore marked as an exception. This is a procedure carried out by administrator consisting of connecting exception CPTs with new context.

Let us consider processing of pump working circuit, which is also made of casting, and can have structure irregularities. If casting structure has irregularities, CPT 'porous' is added to context 'pump manufacturing' (Fig. 2). This CPT is associated with CPT pump circuit. At this moment, a new context needs to be categorized. Based on algorithm for determining context similarities, ASM concludes that new context is similar to context "valve manufacturing exception".

ASM proposes establishing of an association between CPT_X and CPT_N based on algorithm for similarities determination, which categorizes new situation as an exception. After ASM concludes, all data on process and its activities are sent to expert system. If rules that can find a solution to occurring problem are developed within expert system, then those rules are executed. Result of execution of those rules is new process definition, to which are added digging-up and welding. New process definition is then returned to MD system, which continues execution of that process instance, but according to new definition now.

System Architecture of Workflow Management System with ASM

Workflow management system is realized in terms of Java classes in J2EE environment that includes code for system function (Fig. 3). Interface towards these

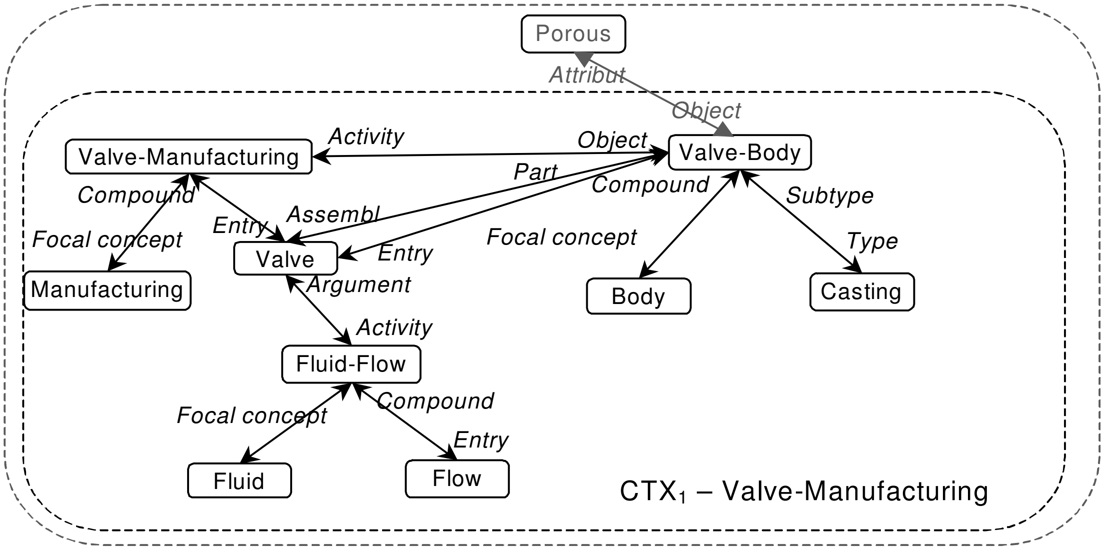

Fig. 1—Context valve-manufacturing-exception

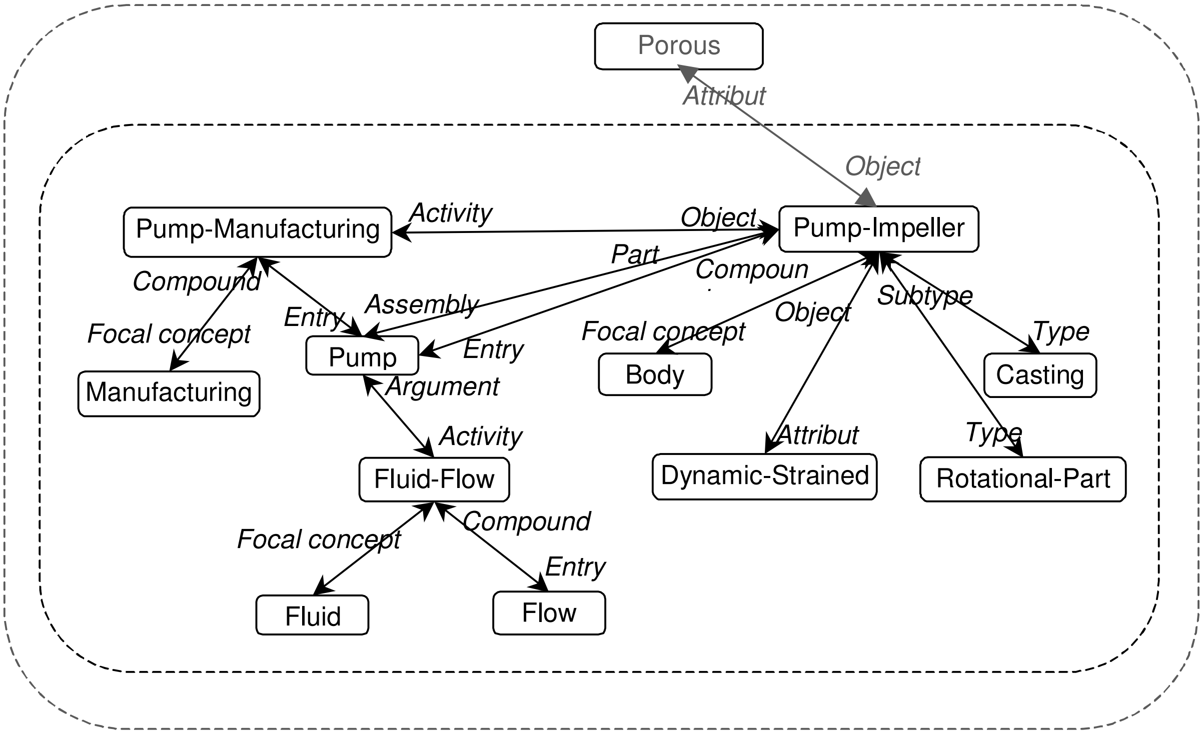

Fig. 2—Context pump-manufacturing-exception

classes can be different. MD system uses web client, that is, sequence of JSP (Java Server Pages) pages. In ASM system, parts are also developed in Java. ASM interface are also JSP pages. Connecting ASM module

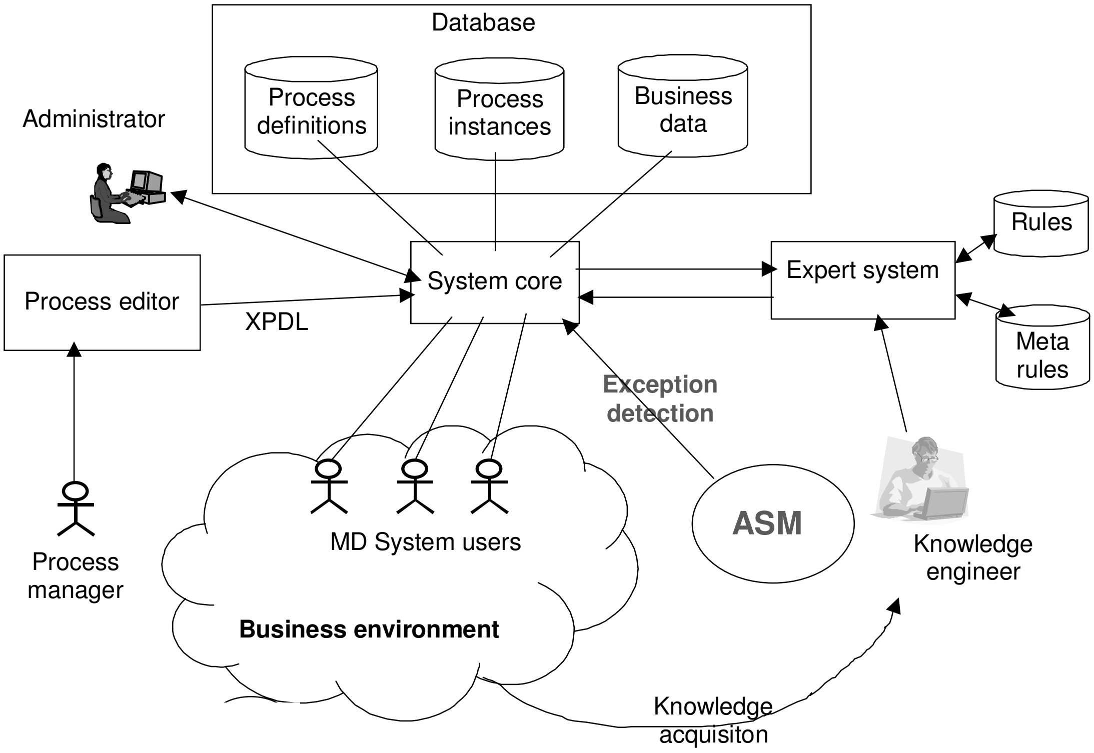

Fig. 3—MD system architecture with exception detection mechanism

is associated with moments of exception appearance (its detection).

Conclusions

This study suggested a new approach to exception detection in process management systems. ASM is employed along with semantic networks, expert systems, object oriented programming and fuzzy logic in J2EE architecture.

Acknowledgment

Part of project was sponsored by Ministry of Science and Technology (project id TR12010).

References

1 WfMC: Workflow Management Coalition, Terminology and Glossary, Document Number WFMC-TC-1011. (1999) http://www.wfmc.org/standards/docs.htm/.

2 Eder J & Liebhart W, Contributions to exception handling in workflow management, in Proc *EDBT Workshop on Workflow Management Systems* (Valencia, Spain) 1998, 3-10.

3 Zongwei L, Sheth A, Kochut K & Miller J, Exception handling in workflow systems, *Appl Intell*, **13** (2000) 125-147.

4 Casati F, Specification and implementation of exceptions in workflow management systems, *ACM Trans Database Syst*, **24** (1999) 405-451.

5 Stojkoviæ M, Maniæ M & Trajanoviæ M, On semantics of design and manufacturing features in digital product model, in *Proc 2nd Int Conf on Manufacturing Engineering ICMEN*, (Kallithea, Greece), 2005, 731-740.

6 Stojkovic M, Manic M, Trajanovic M & Korunovic N, Semantic structures in the product data model, in *Proc Int Conf on Product Lifecycle Management, PLM-SP3*, (Inderscience Enterprises Limited, Switzerland), 2007, 227-234.

7 Stojkoviæ M, Maniæ M, Trifunoviæ M, & Mišiæ D, Recognition of design features of cad model by its semantic features analysis, in Proc CD Yu-Info Conf, (Information society of Serbia, Kopaonik, Serbia) 2009.

8 Enhydra Shark: Open Source Java/XML Workflow Engine, http://shark.enhydra.org/

9 Borgida A & Murata T, *A unified framework for exceptions in workflow and process models – an approach based on persistent objects*, **Technical Report** (Rutgers University, Newark, New Jersey, USA) 1998.

10 Borgida A & Murata T, Tolerating exceptions in workflows: a unified framework for data and processes, in Proc Int joint Conf on Work Activities Coordination And Collaboration, (ACM, California, USA) 1999, 59-68.

11 Song Y & Han D, Exception specification and handling in workflow systems, in *Web Technologies and Applications, Lecture Notes in Computer Science* (Springer, Berlin / Heidelberg) 2003, 594.

12 Hwang S & Tang J, Consulting past exceptions to facilitate workflow exception handling, *Decision Support Syst*, **37** (2004) 49-69.

13 Weber B, Wild W & Breu R, CBRFlow: Enabling adaptive workflow management through conversational case-based reasoning, in *Advances in Case-Based Reasoning, Lecture Notes in Computer Science* (Springer, Berlin / Heidelberg) 2004, 434-448.

14 Shaofei W & Guo B, Design of an exception handling algorithm of workflow, in *ISECS Int Colloquium on Computing, Communication, Control, and Management*, vol 3, (IEEE Computer Society, Washington, DC, USA), 2008, 281-284.

15 Weber B, Rinderle S & Reichert M, *Change Support in Process-Aware Information Systems - A Pattern-Based Analysis, Technical Report, TR-CTIT-07-76, CTIT* (University of Twente, The Netherlands) 2007.