FACTA UNIVERSITATIS
Series: **Mechanical Engineering**
https://doi.org/10.22190/FUME211115026M

Original scientific paper

DETECTION AND HANDLING EXCEPTIONS IN BUSINESS PROCESS MANAGEMENT SYSTEMS USING ACTIVE SEMANTIC MODEL

Dragan Mišić, Miloš Stojković, Milan Trifunović, Nikola Vitković

Faculty of Mechanical Engineering, University of Niš, Serbia

Abstract. *Although business process management systems (BPM) have been used over the years, their performance in unpredicted situations has not been adequately solved. In these cases, it is common to request user assistance or invoke predefined procedures. In this paper, we propose using the Active Semantic Model (ASM) to detect and handle exceptions. This is a specifically developed semantic network model for modeling of semantic features of the business processes. ASM is capable of classifying new situations based on their similarities with existing ones. Within BPM systems this is then used to classify new situations as exceptions and to handle the exceptions by changing the process based on ASM's previous experience. This enables automatic detection and handling of exceptions which significantly improves the performance of bpm systems.*

Key Words: *Business Process Management Systems, Exception Detection, Exception Handling, Active Semantic Model, Analogy-based Reasoning*

1. INTRODUCTION

Business process management systems (BPMS) are software systems which manage business processes. So far, these systems have been proven useful for management of the processes with a solid structure, in which changes do not occur often. On the other hand, BPMS are also used in environments in which there is a constant need for deviation from predefined process (e.g., logistics, healthcare). In the terminology used for BPMS, deviations from the predefined model are called exceptions.

Exceptions can be anticipated in which case the issue is handled by incorporating it in the model at the process modeling time. This approach leads to the creation of very

*Received: November 15, 2021 / Accepted May 17, 2022
Corresponding author: Dragan Misic
Faculty of Mechanical Engineering, University of Niš, Aleksandra Medvedeva 14, 18 000 Niš, Serbia
E-mail: dragan.misic@masfak.ni.ac.rs

complex models with many branches describing alternative pathways in the case an exception occurs. Nowadays, this approach is almost completely abandoned.

Unanticipated exceptions are handled at the process execution time. In that case, it is necessary to have a way of changing the business process and adjusting it to the new circumstances. This approach is used in most modern BPMS [1].

Exception handling in BPM systems is of great importance for the successful process execution. Research shows that the management of process exceptions requires a lot of resources, it costs a lot [2], its success is critical [3] and it is time-consuming.

One of the possible ways of making a system capable of recognizing and categorizing the unknown situation as an exception is by enabling recognition of similarities and differences between the semantic features of an unpredicted situation and the known ones previously semantically interpreted and categorized as an exception. That is exactly how the Active Semantic Model (ASM) functions, and we use it in this paper for detection and handling of the exceptions. In the early stage of its use, this system is able to assist people to solve problems. With time, it collects knowledge and it becomes capable of offering intelligent advice and proposing solutions independently.

The main contribution of our work is the use of ASM for presenting the knowledge and making conclusions based on that knowledge. ASM may independently offer a solution to the problem that occurred and recommend the process adaptation accordingly. In that way we reduce the need for direct human involvement in handling exceptions, because after the training, ASM is capable of handling challenging situations independently. ASM is more flexible than other methods in artificial intelligence such as case-based reasoning or ontologies, because ASM is able to reuse the events from other domains to make decisions in a new domain. Additionally, the formalisms do not need to be defined in advance, unlike with the ontologies.

This paper represents a sequel to our work on this subject which is presented in [4] and [5]. In [4], we describe how we expanded the process model defined in the XPDL (XML Process Definition Language) language by adding constructions which refer to assignment of resource to activities. In [5] we describe how ASM is used for detecting exceptions. In this paper, we show that ASM is now able to solve problems (it offers a solution to a certain situation), and this we applied to the problems which occur due to inadequate resources. Its ability to reach conclusions is also improved since we have significantly improved the algorithm based on which ASM recognizes the topological similarity it uses for handling exceptions. ASM is now capable of making meaningful judgments, conclusions and decisions in new and unexpected situations. Compared to other approaches to semantic modeling (e.g. ontologies), ASM has an autonomous, flexible and significantly more analytical mechanism of semantic categorization of data.

The rest of this paper is organized as follows: In Section 2 we present some of the relevant papers in this area. In Section 3 we show the ASM structure and its reasoning methods. Section 4 explains how ASM is connected to BPMS and how it can assist with detection and handling of exceptions. The results of the ASM conclusions are explained in detail in section 5. Application of ASM in the BPM systems is discussed in Section 6. The paper is concluded in Section 6.

2. Related Work

As mentioned earlier, the exceptions during the execution of business processes are frequent in practice. Handling these exceptions is therefore significant for organizations employing BPMS systems. For instance, in [6] the authors analyzed the relations between the occurrence of exceptions and operational performance. Their research indicates that the exceptions lead to poorer operational performance: the processes where the exceptions occur take longer to complete than the processes with no exceptions, underlying the importance of the BPMS systems that can adapt to changes.

In [7], the authors performed the analysis of existing process management systems with respect to their support for flexible, emergent and collaborative processes. They conclude that the contemporary systems do not adequately support these three process characteristics.

In [8], the authors work on supporting users at the inflection point. The inflection point is a place in the process execution where an unforeseeable eventuality arises. At that moment, the mechanism that gives recommendations about adaptation to new circumstances is launched. This is done based on the search of the existing workflow specifications, which are located in the repository.

Examination of the previous cases is also used by the authors who apply the Case-Based Reasoning [9] techniques in order to identify and solve exceptions. For example, in [10], authors use so-called adaptation cases. The adaptation cases describe situation-specific adaptation traces, which can be transferred to another, similar situation and replayed there. Based on defined adaptation cases, it is possible to apply the adaptation which was used earlier to a new situation, too. In [11], the authors suggest using the Conversation CBR techniques [12] to update the process. The authors extend the basic CBR mechanism with automatic question creation technique that leads the user through describing the new process. The questions are created based on the analysis of existing processes.

The model used in the ADEPT system [13] uses an ad-hoc approach. The processes are described *via* specific language in which there are operators for dynamic insertion, deletion or transfer of activities during the process execution. The disadvantage is primarily the fact that there are no algorithms that will automatically determine the circumstances under which it will apply a specific workflow adaptation.

In the paper [14], the authors state that the existing mechanism for handling exceptions in business process execution language is not completely satisfactory. The main disadvantage lies in the fact that the behavior of the system in the cases when the exception occurs must be defined in design time.

In [15] and [16], the authors describe the application of ontologies to enhance the flexibility of BPM systems. They allow for defining ad-hoc activities followed by the decision which process to run based on ontologies [15]. In [16], the ontologies help define advice for users when creating new processes. The processes are offered to the user only if they are in accordance with the rules which are defined within the ontologies.

In agile BPM [1], the reactive rule model is utilized to recognize exceptional circumstances automatically and to determine the necessary process instance flow adaptations. For this purpose, failures trigger new obligations, which are the principal motivators for agents to act. Based on obligations, agents can dynamically replace/re-plan the failed goal, trigger a repair action, or abort/roll-back the execution.

For process monitoring, detection of unanticipated exceptions and automated resolution, the Cognitive Process Management System can be used [17]. This system relies on the technologies from the field of knowledge representation and reasoning. For modeling the primary domain where the processes are run, situation calculus is used; for structure specification and control flow of the process, it uses the IndiGolog (agent programming language) while for process adaptation it uses automatic planning.

As can be seen, there are many different approaches to handling exceptions in BPMS. Some of these approaches only help people with solving problems. Others attempt to offer a certain level of automation, i.e. the use of previously defined solutions.

Our opinion is that the level of automation in handling exceptions may be raised if the knowledge about the problem and the process is presented in a way computer can easily use it. As mentioned before, ASM may offer a solution to the problem that occurred and recommend the process adaptation accordingly. This is not the case with the approaches which represent the tools that help humans with the system adaptation [13]. ASM is an Analogy-based Reasoning technique (ABR), and so are the aforementioned Case-based reasoning techniques [10, 11]. In comparison with these techniques, ASM offers better solutions because it is not limited exclusively to the solutions which come from the same domain.

When compared with the approaches that achieve flexibility by using the ontologies [15,16], it can be said that our system overcomes some of the main problems which exist in ontologies, such as that the semantic reasoners designed to work with DL-founded ontologies showed themselves weak in making relevant entailments beyond the predefined and embedded logical formalism of deduction. Similar is also true for reasoning flexibility – ability to make relevant, but quite different entailments about the same concept for semantically distant or different contexts (vocabularies) with a single set of logical inference rules and axioms on disposal. Finally, having analytical ability to autonomously dissolve a portion of knowledge about one concept or group of concepts from one context and apply it to a quite different (semantically distant) concept or a group of concepts that are inherent to equally different context is something which appears not as a strong side of the richly axiomatized ontologies which rely on first-principles reasoning approach.

3. ACTIVE SEMANTIC MODEL

ASM is a specific model of the semantic network that was originally developed in-house and is described in more detail in previous papers [18, 19, 20]. The specificity of this model originates from its feature that the meaning (semantics) of a certain concept (which is usually represented as a node in a semantic network) is defined not in its attributes, but in the attributes of its relations to other concepts (The term *association* is favored instead of relation since this structure helps the ASM algorithm to associate or point out correspondent inference i.e., semantic categorization). Every single association is featured by eleven attributes where two of them are tags (*names*) of concepts this relation associates: cpt_i, cpt_j. Since these attributes – tags of concepts, can exist in more than one relation, they are a kind of junction of these associations, and in this way, they can be considered as virtual nodes of semantic network. Beside two different *concepts*, every single association is defined by additional three groups of attributes: topological (*roles* (r_i, r_j), *type* (t), *direction* (d), *character* (c)), weight (*accuracy* (h), *significance* (s)) and affiliation (*context id, instructor id* or *user id*). This kind of associations' structure enables application of an original algorithm for efficient recognition of similarities between network's sub-graphs or network fragments. (The term *plexus* of associations is preferred instead of associations' sub-graph due to its feature to connect the concepts from different contexts, hence, not just in one layer, i.e., not just in a graph-plane). This algorithm drives analogy-based reasoning process in the core of the model's inferring engine. The ability to determine the type and degree of similarity between sub-graphs of the network makes the inferring engine extraordinary autonomous, flexible and analytical in data semantics interpretation. These features are especially important in the cases of unpredicted inputs and small or incomplete networks [18, 19, 20].

3.1 Communication between the user and ASM

The most usual case of communication between the user and ASM is being performed by entering the new concepts into the semantic network. This is performed by forming the new associations that include the new concept. By associating the new concept with other concepts that exist already in the ASM semantic network, the inferring engine of ASM is being triggered immediately. This results in proposing (creating) additional new associations between the new concept and other concepts and contexts in the network by ASM itself autonomously. Each new association is an elementary piece of knowledge that enables further semantic categorization of a new concept. The user can correct the attributes of the associations that are proposed by ASM or remove the proposed associations; thus the user corrects its semantic categorization i.e., the way how ASM infers. In addition, by correcting it, the user keeps improving ASM for future autonomous analogy-based reasoning. Hence, while associating a new concept into its network, ASM enlarges its semantic network gaining a new piece of knowledge in addition to improving, at the same time, the algorithms for analogy-based reasoning. Also, by proposing the new associations autonomously by employing ABR, ASM provides new semantic categorizations of a new concept or a new context, which are actually a kind of intelligent responses that the user expects from ASM. So, ASM always works in both regimes - acquiring the knowledge and providing the intelligent inferences at the same time. Fig. 1 shows an example of how ASM learns and infers simultaneously. After the user forms a few new associations that connect one or several new concepts with the rest of ASM's semantic network, i.e., to the other several concepts, which exist already in the ASM's semantic network, building an input association plexus PLX_X in this way, ASM, firstly, scans the network looking for a set of association plexuses $\{PLX_N\}$ which are topologically analogous to the input association plexus PLX_X in which the new concept CPT_1 appears: $PLX_X \asymp (PLX_N)$ (Fig. 1). Actually, ASM recognizes fragments (sub-graphs or plexuses of associations) of more complex structures that exist in the semantic network of ASM (e.g., $PLX_N \leftarrow Frg(CTX_N)$) which are topologically analogous to the input plexus. Once it recognizes the topological analogy between the input association plexus PLX_X (which is new to ASM) and existing association plexuses $\{PLX_N\}$, a procedure for upgrading the input association plexus PLX_X is triggered and performed according to the model of the existing association plexuses $\{PLX_N\}$ that are topologically analogous to PLX_X. The upgrading of PLX_X is performed by creating new associations between the "known" concepts that exist in ASM's semantic network, and "unknown" concepts (that are included in PLX_X). These new associations are being created by ASM autonomously. For example, ASM reacts by proposing creation of a new association $A^X_{1,4}$ between concepts CPT_1 and CPT_4, emulating association $A^N_{11,14}$ (between concepts CPT_{11} and CPT_{14}) from topologically analogous plexus PLX_N which is a fragment of context CTX_N. The new association will have the same topological parameters as association $A^N_{11,14}$. In that way ASM categorizes the new concepts semantically, in other words, forms their meaning in the new (current) context; these new associations are practically the resulting conclusions about them. Various algorithms which ASM uses for reasoning are explained in more details in [18, 19, 20].

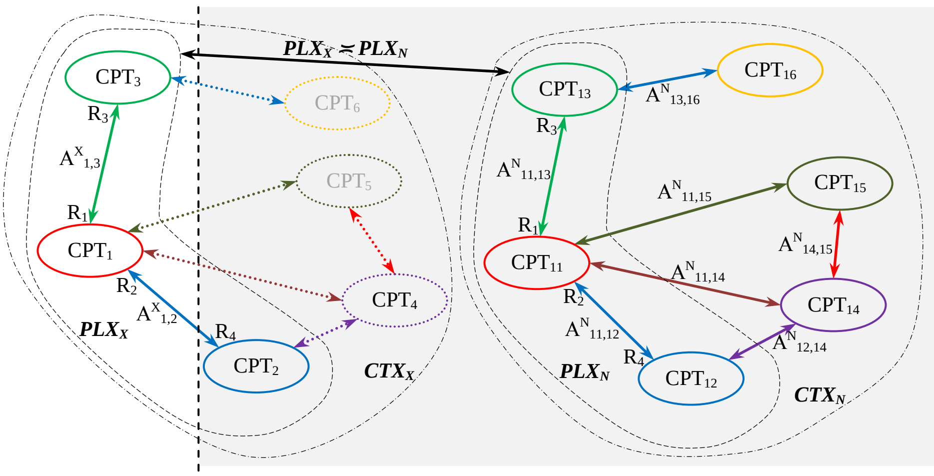

Fig. 1 Topologically analogous association plexuses: $PLX_X \asymp PLX_N$

4. Connecting ASM and BPM Systems

The process model consists of activities which are executed in a specific order. This model is later used for the creation of specific instances, which correspond to the real processes. For process management we used the MD system, developed at the Faculty of Mechanical Engineering in Niš [4]. It is based on the Enhydra Shark engine [21]. For the model definition and process execution in the Enhydra Shark system, the XPDL is used.

XPDL is a standard XML based format defined by the Workflow Management Coalition (WfMC). Its aim is to enable the exchange of process definitions between various tools used to create processes. Enhydra Shark uses XPDL not only for data exchange with other systems, but also as the main way of representing processes within the system.

Enhydra Shark is not able to handle process' exceptions and respond appropriately in the situations when the process deviates from the model that is predefined. Such situations are resolved by ASM and the expert system [5]. Solving problems can lead to changing individual activities, but also to the changing of the entire process and its definition.

Illustration of the connection between the MD system and ASM is shown in Fig. 2.

In the first version of the MD system for handling exceptions, an expert system was used with the expert shell JESS. The core of the MD system was written in Java that matched the expert shell JESS and made it relatively simple to connect the expert system with the process management system.

The process in the MD system is described by one definition and multiple instances made based on that definition. The task of the expert system is to help with detecting exceptions that may arise during the process execution as well as to propose a solution. The solution often consists of a proposed change to the process. The changes can refer only to the current process instance, but also to all other instances created from the same process definition. This is defined within the rules of the expert system that update the process upon their execution. The rules are defined for specific processes. If a new process should be monitored, then it is necessary to define suitable rules that will handle the exceptions that may arise in the new process. The procedure that performs this task is defined by a set of functions written in Java and Jess script language that are later invoked from the action part of the rule [4].

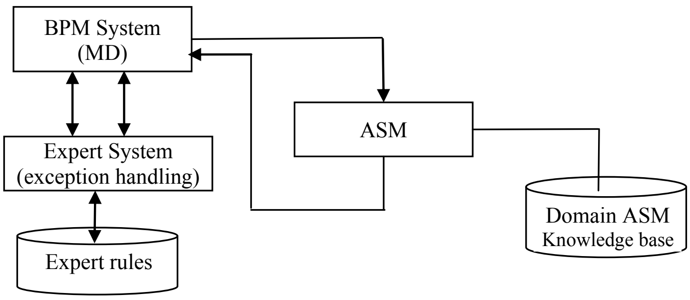

Fig. 2 MD system with ASM

Exception detection and handling using an expert system are limited by the rules that are defined in the system. In an attempt to overcome this limitation, the MD system is connected to ASM that is capable of making conclusions based on analogies. This connection improves the quality of the MD system by significantly enhancing the system's capabilities for automatic detection and handling of exceptions. Without ASM, it was possible to detect exceptions from signaling that a resource is missing based on the values of control parameters and whether the execution time was over the time limit of a certain activity. ASM enables the system to consider the big picture and connect the situations that were previously labeled as exceptions and took places in completely different processes, with the current situations. ASM is also able to offer a solution based on the analogy with some of the previous situations.

The application of ASM for detecting and handling exceptions will be explained in more detail using the orthopedic implant design and manufacturing process as an example. The outputs of this process are the orthopedic implants adapted for the patient. The process is managed by the previously mentioned MD system.

The preparation of an orthopedic implant includes the preparation of osteofixational material comprised of the scaffold and the fixator. The scaffold is a piece of the bone implant assembly (entitled Ossification Material in figures), whose main functions are to substitute the missing part of the bone tissue and to hold the bone graft inside the volume of the scaffold during the tissue recovery process. This allows the communication with neighboring tissues. Fixator is another piece of the bone implant assembly, which should fix (fasten) traumatized parts of the bone into regular anatomical position and transfer a part of the load that bone bears while organism is trying to heal the bone, that is, generate a missing piece of the tissue. The proto-tissue or bone graft (entitled "Bone Part" in figures) is the third piece of the compound bone implant assembly that usually consists of fat tissue, stem and/or progenitor cells and other soft and liquid substances.

In order to enable ASM to perform the process analysis, it is necessary to semantically describe the process and its elements. That means that it is necessary to present all the essential elements of the process by using the structures from ASM.

The semantic description of the process elements is done by the system administrator at the time of initiation of the first process instances, and it is based on the process definition. Defining of concepts and associations between them is done by using a graphical editor. Initially, the concepts which represent data from XPDL process definition are displayed in the editor. It is up to the administrator to accurately describe the process, by which he improves the quality of later ASM's conclusions. That means the administrator has a role of ASM's instructor. An example of an ASM context for the process which manages the process of preparing and manufacturing the osteofixation material (OM) is shown in Fig. 3.

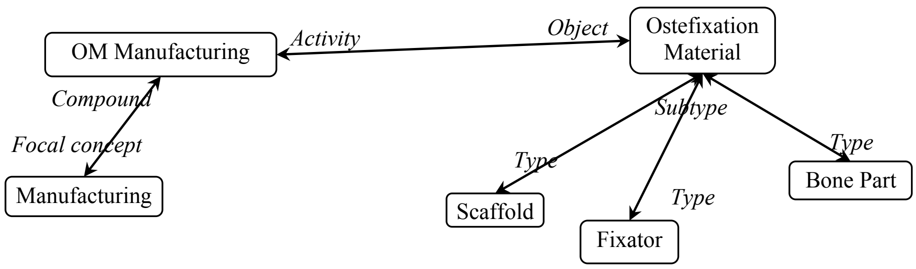

Fig. 3 ASM context for Osteofixation Material Preparing and Manufacturing

Each process activity is represented by a particular plexus of associations in the ASM. These plexuses also contain descriptions of the data required for the execution of these activities as well as the data made while executing the activity. The data which describes the activity is both defined by the administrator and taken from XPDL process definition. There is also information about the resources that are required for the normal execution of the activities. Most importantly, this may include the material resources because people who perform these activities are represented by separate XPDL elements (participant element).

An example of defining the activity within ASM (scaffold modeling from Fig. 6) is shown in Fig. 4.

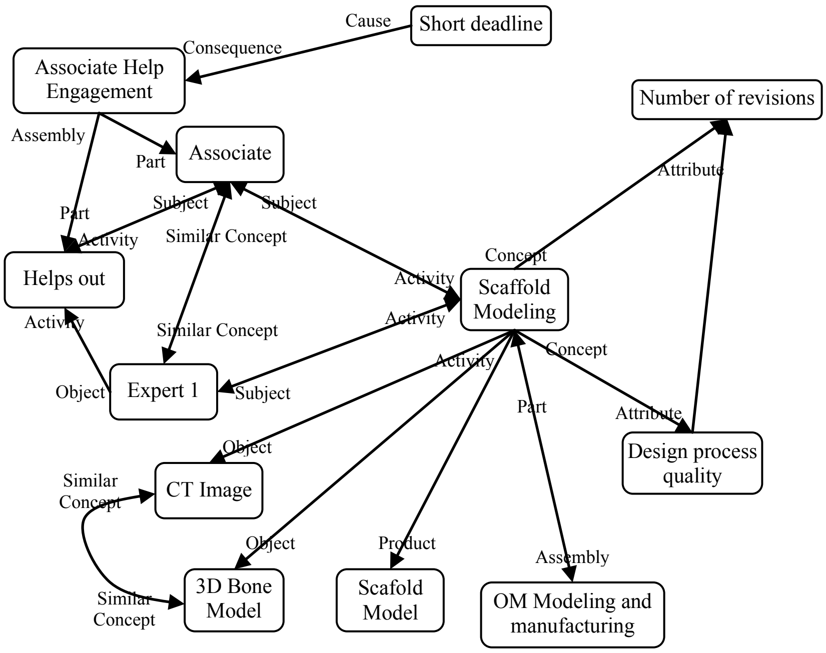

Fig. 4 Association plexus (context) for the model of the activity Scaffold Modeling

It should be mentioned that the first step is to give the description of the activity model to ASM. The contexts of specific activity instances represent subtypes of the initial context. The association between these contexts is shown in Fig. 5.

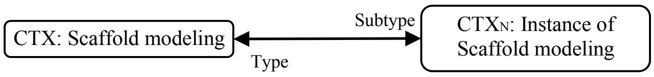

Fig. 5 Connection between the context of activity model (CTX) and the context of specific instances (CTX_N)

If an exception occurs during the process execution, the first issue is to recognize the new situation as an exception. As we mentioned before, activities and their execution environment are described by the semantic network which also contains data that the activity uses. The new situation is usually manifested through the data that characterize a process. What usually happens in such situations is that certain new data occur or that the existing data receive some special values. In the system’s learning phase, the instructor should characterize the new situation as an exception and offer a solution to it by introducing the association between the concept exception and that solution. In the application phase, ASM should recognize an analogy between the new situation and what it has learned, and independently propose the qualification of a new situation as an exception at first besides offering the solution.

5. Process of Implants Design and Manufacturing as an Example of the ASM Based Reasoning

We will show how ASM draws conclusions on the example of managing processes that may occur in the same company. Let us suppose that there is a process of designing and manufacturing osteofixation material (scaffold and fixator) which is adapted to the patient. The process begins in a hospital, when a patient comes to the doctor with a fracture which is to be treated. The first thing the doctor should do is to define the type of the treatment which the patient will undergo. He makes that decision based on radiology image. If there are no parts of bone missing, a fixator will be set and it will allow the bone to heal properly. If a small part of bone is missing, it is needed to design and manufacture a scaffold, which is filled with cellular material that will allow the missing bone part to regenerate. It may happen that it is needed to set a fixator, in addition to the scaffold. The third possibility occurs when a large part of bone is missing; in this case it is needed to make a fixator as well as the missing bone part.

After making the decision about the treatment, the process is continued, part in the hospital, part in the company which manufactures osteofixation material. If the patient needs a scaffold, the first step is to create a parametric model of the bone and the scaffold from the parameters determined by the doctor. Using that model, the manufacturer creates the scaffold and designs and constructs the fixator. The scaffold and the fixator are then sent to the hospital where the surgeon will use them for the operation. The process diagram defined inside the MD system is shown in Fig. 6.

5.1 Training an ASM

The activity of this process which we are interested in is scaffold modeling. The scaffold is geometrically very complex, so its modeling is not simple (it is a kind of armature needed for the bone recovery of a specific patient). The proper modeling of the scaffold requires time as well as the extensive experience of the engineers.

In the operation of the scaffold modeling the engineer creates so-called scaffold struts connecting nodes one by one and puts them on the surface. This structure differs for each patient. Cross-section, intersection angle and density of the scaffold struts may change depending on desired mechanical strength of the scaffold. The process consists of iterative sequences. The accelerated work of the engineers may easily be the cause of the relocation of the connecting points of the scaffold struts or some other mistakes while modeling, which leads to problems with the structure of the scaffold modeled in such manner.

This activity is followed by control activity (activity Model Control from Fig. 6). In the case that there is something wrong with the design, the model is returned for revision. If the number of revisions is excessively increased (e.g. more than five), we conclude this is a sign that there is something wrong with the modeling, and that certain steps must be taken. The reaction to such situation is anticipated and embedded in the system (as an if-then procedure, which is a part of the process).

Also, there are defined deadlines for scaffold modeling, which depend on the patient's injury and the urgency of the surgery.

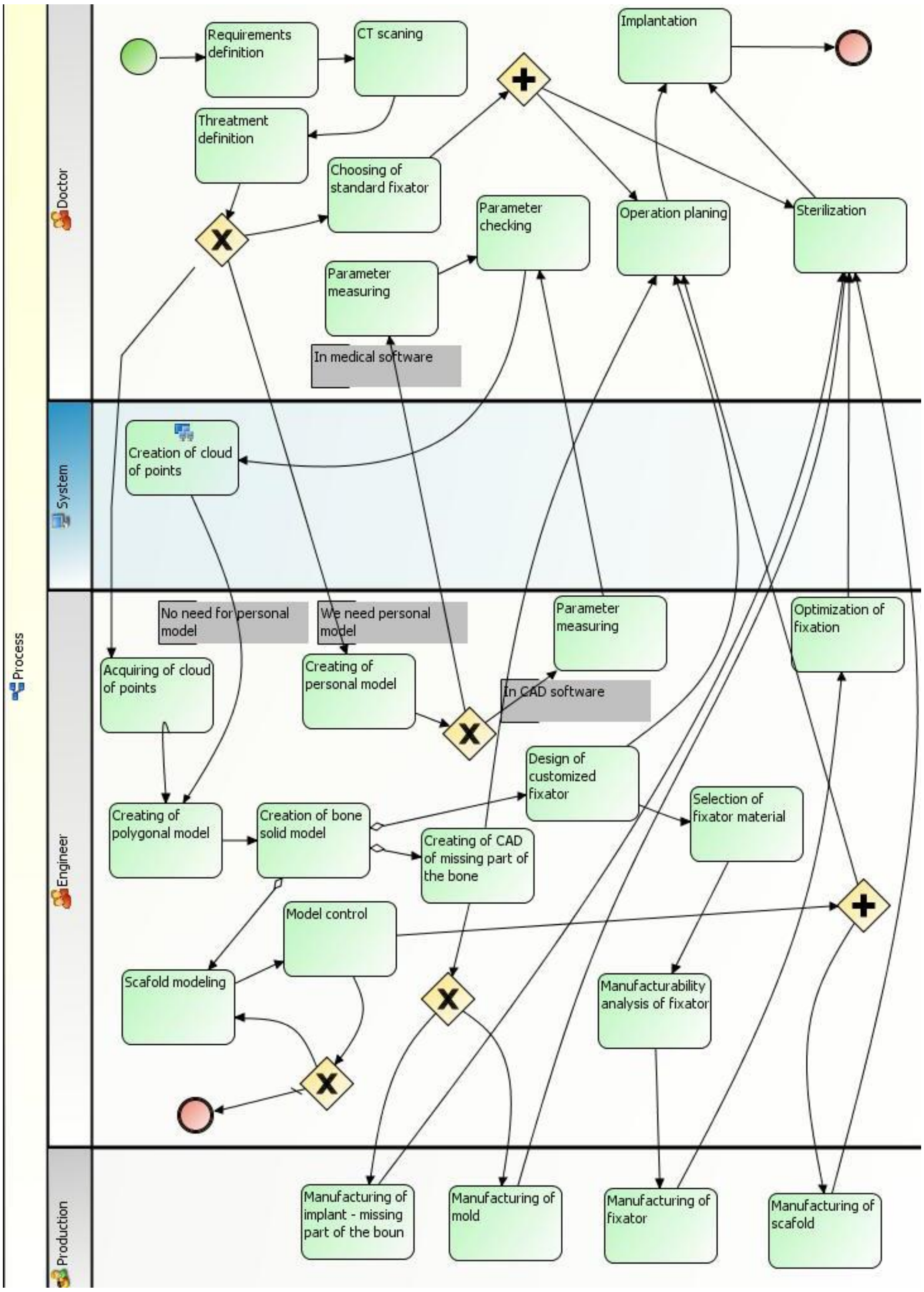

Fig. 6 Process for ostefixation material preparing and manufacturing (process diagram created within the MD system)

Occasionally it happens that the deadline for manufacturing of osteofixation material is very short. Therefore, the deadline for scaffold modeling is also very short. ASM will be notified of that by adding a new concept – Short Deadline (Fig. 4).

Short deadline will make an engineer hurry up with the model design, which may cause an increased number of mistakes. In order to preserve the quality of the model and prevent the bottleneck from occurring in this activity, the first predefined reaction is to engage an additional expert. ASM association plexus for this activity at model level is modeled as shown in Fig. 4.

However, it sometimes happens that the deadline is missed, despite the engagement of the additional expert. This is the case when the number of revisions stayed below the specified limit (e.g. 5), so the embedded procedure for the case of an excessively increased number of revisions was not launched. ASM is notified of this by adding the concept Small Number of Revisions which is an attribute of the concept Number of Revisions. This refers to a specific activity instance (Fig. 5).

Missing a deadline is an exception for BPMS. In cooperation with the engineers involved in the process, the system administrator is documenting that a short deadline may cause the operation to fail, despite the engagement of an additional expert. In such cases, the number of revisions stayed small. Therefore, the situation which is characterized both by a small number of revisions and short deadline may lead to missing the deadline.

There are two ways of solving this problem. The first one is to embed an if-then procedure in BPMS, and that procedure would be executed in the case when such (numerically expressed) short deadline and the number of model revisions occur. It should be noted that this procedure can be applied only if the same situation occurs again in the same context. Such formalized knowledge, however, is impossible to apply to a case from a different domain. It is also impossible to apply it to a case from the same domain if the conditional parts do not completely match.

In order to enable the acquired experience to be applied to other domains, the administrator can provide ASM with new information. Thus, new associations are manually added to ASM by the administrator (Fig. 7). These associations are added to the context which refers to a specific instance of the activity Scaffold Modeling (CTXN). The first association connects the concept Short Deadline with the concept Unsuccessful Operation and, therefore, defines it as a cause of operation failure. The second cause of an unsuccessful operation is a small number of revisions. This is represented by the association between the concept Small Number of Revisions and the concept Unsuccessful Operation.

Based on experience and conversations with the engineers, the system administrator reached the decision to set the accuracy of the first association to 50%. By doing this, he wanted to highlight that there is a 50% probability that a short deadline will cause the operation to fail. It is also estimated that in the given context, this association is very significant, so the association significance is set to 75% or 100%. The same parameters are set for the associations which are related to the concept Small Number of Revisions and Unsuccessful Operation.

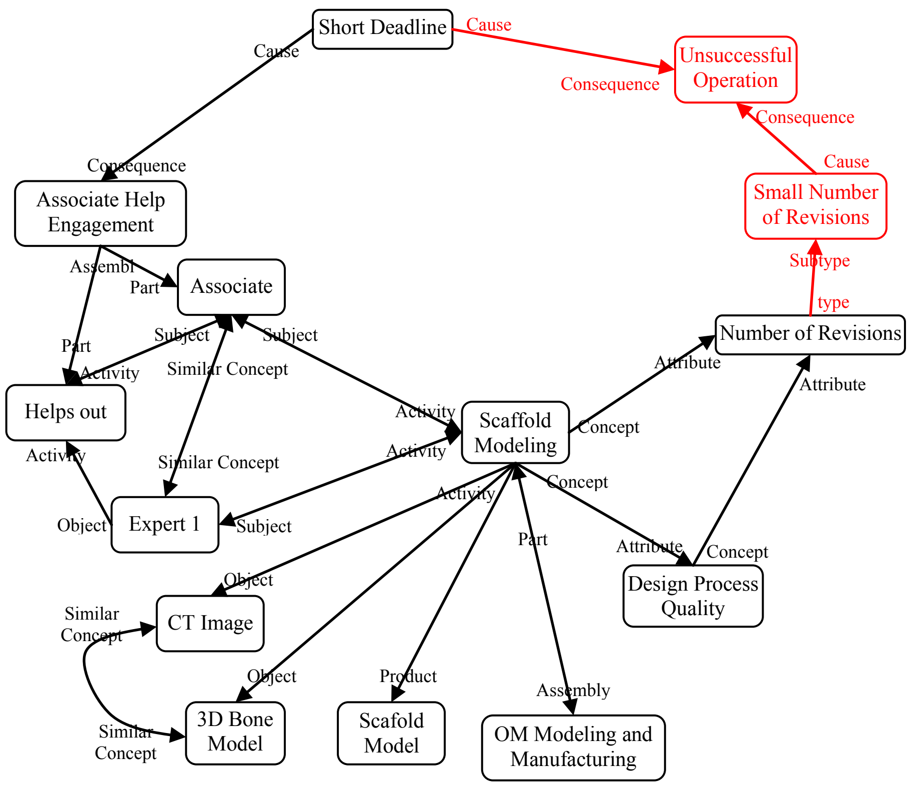

Fig. 7 Association plexus (context) for the activity Scaffold Modeling with added associations (at instance level)

The altered context of the activity Scaffold Modeling represents an exception. ASM is notified of that by adding a new association between the context and the concept Exception (Fig. 8). Scaffold Modeling activity plexus in Fig. 8 represents the whole context shown in Fig. 7 (it refers to a specific activity instance – CTX_N).

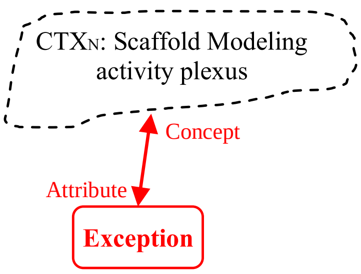

Fig. 8 Categorizing the situation as an exception

In addition to documenting the new situation by adding new associations, the system administrator, in cooperation with the engineers, has considered the ways of overcoming such situations in the future. For such cases, it could be useful to consider the application of a specific designing method characterized by applying so-called UDFs (User Defined Features). That approach, which involves usage of partially pre-defined geometric forms, can accelerate the process of modeling, and simultaneously decrease the number of model revisions. In the case of bone scaffold design, UDFs could be pre-defined forms of scaffold's struts and connecting nodes.

This conclusion leads to the decomposition of the Scaffold Modeling activity into two activities. During the first activity, the UDFs would be prepared, and in the second activity the scaffold would be designed using the prepared elements (UDFs). The second activity is now performed much faster because it is needed only to define positioning references and dimension parameters for each UDF. Using of UDFs for designing complex geometric forms could be semantically interpreted as a subtype of some more general activity, which can be e.g. entitled as Sequential Job Decomposition. This relation should also be taken into account that is embedded into the ASM network.

ASM is notified of the conclusions made by the administrator and the engineers by adding the new associations to the system, which connect the concept Exception with the concept Alternate, which is further connected with the concept UDF Based Scaffold Modeling which is a subtype of the Sequential Job Decomposition (Fig. 9).

The accuracy of the association which indicates a possible solution is 50%. This describes the fact that the sequential job decomposition is not the only possible solution.

The offered solution may be permanently applied to this process, so the ASM administrator will teach ASM by associating this solution with the general context of the Scaffold Modeling activity – CTX (Fig. 9), thus signalizing that the process should be permanently changed.

5.2 Semantic categorization of a new process

In some new situations that are more or less similar to the previous ones, ASM can now apply what it has learned from these previous situations.

The process of recognizing an unpredicted exception and its categorization is performed by comparing the similarities of the context describing the current situation (activity) with the already existing contexts. In this case, the contexts of new activities are compared to the context of the Scaffold Modeling activity. The comparison of the context similarity according to the content is based on the similarity of plexus topology (a kind of subgraph isomorphism) [4, 18, 19]. In accordance with the topological similarity (difference) which it recognizes between these contexts, ASM will semantically categorize the new situation in regard to the existing situations. If a new situation is similar to the situation categorized as an exception, then it is suggested that the new situation should also be categorized as an exception, with the calculated/assessed magnitude of the assertion accuracy.

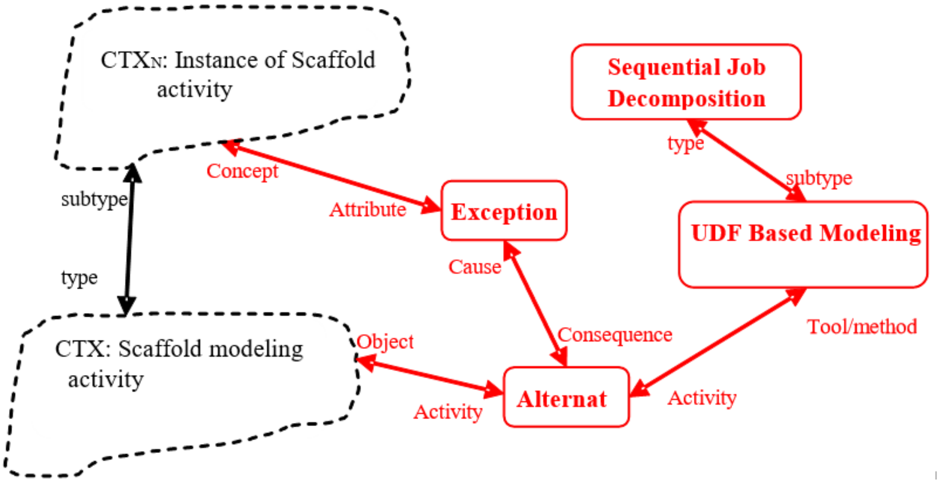

Fig. 9 The associations which define the problem solution

After an exception is detected, ASM will try to offer a solution for the occurring problem. The procedure is similar to the one which is used when an exception is being detected. ASM compares the similarity of the plexus of associations which describe an exception, to the plexuses which exist in the ASM network and which are described an exception. If the ASM discovers that there was a solution to the problem in any of the predefined plexuses, it will offer such a solution to the new situation as well.

The process which we used as an example of the application of the knowledge implemented in the ASM refers to the manufacturing of customized hip endoprosthesis. The process model is shown in Fig. 10 (due to complexity of presenting the whole process, only the part of the process relevant for the paper theme is shown).

Hip endoprostheses consists of three elements. Those are femoral head, femoral neck and femoral body insert. During the process execution, the adaptation of parametric model of all three elements to a specific patient is done based on CT image, after which those elements are manufactured. After manufacturing, the elements are put together and inserted into the patient.

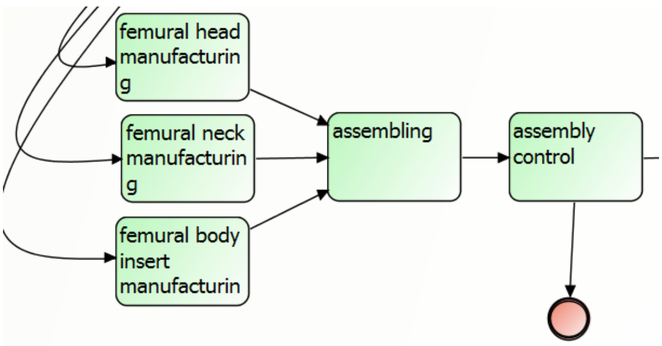

Fig. 10 Part of process for hip endoprosthesis designing and manufacturing

At the beginning of the process, the administrator defines the ASM model for the Hip endoprosthesis manufacturing process (Fig. 11).

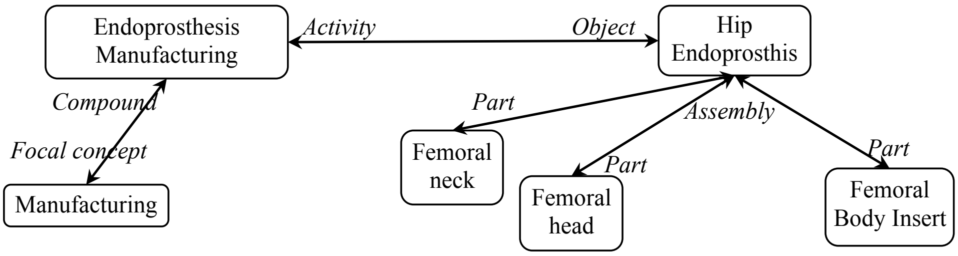

Fig. 11 ASM context for Process Hip Endoprosthesis Manufacturing

The activity of the process we are interested in is the activity Endoprosthesis Assembling. In this activity, the operator initially puts the elements that are completed in the clamping tools, after which the elements are being attached to form one unit. The operator uses a specially designed jig to position the parts accurately. This custom-made positioning mechanism is considered as the main production means for this operation, though there is also an additional tool (means), which may also be used for assembling if needed.

The geometric accuracy of the assembly is determined regarding the angular deviation of so-called femoral neck shaft angle (NSA) from predefined value. The neck shaft angle is an angle between the femoral neck axis and the femoral corpus axis. Since the value of this angle differs for each patient, this deviation is expressed as a percentage, and must not be larger than, for example, 3%. If the deviation is larger than 3%, the process is stopped. After that, the engineer will search for the problem causes and try to eliminate them. This is a formalism implemented in the process as an if-then procedure. Ordinarily, the geometric accuracy of the manufactured assembly is below the required one due to fast manipulation, but it may occur for some other reasons as well.

The context, i.e., plexus of associations that semantically describes the Assembling activity is shown in Fig. 12. This context models the general concept of this activity (CTX). The previous remark that the context which describes the instance of activity is a subtype of the context which describes the model of activity is valid in this case, too.

A sudden requirement for a larger than usual production batch may lead to the acceleration of the assembling process, which, in its turn, may further lead to an increased deviation from the required geometric accuracy of the produced assemblies. The case when this deviation exceeds the allowed limit is covered by a specific if-then procedure. If the angular deviation remains below the allowed value, this procedure is not being launched.

In the following section we will explain in detail the manner in which ASM makes conclusions about a new situation using the knowledge about a familiar situation which once occurred.

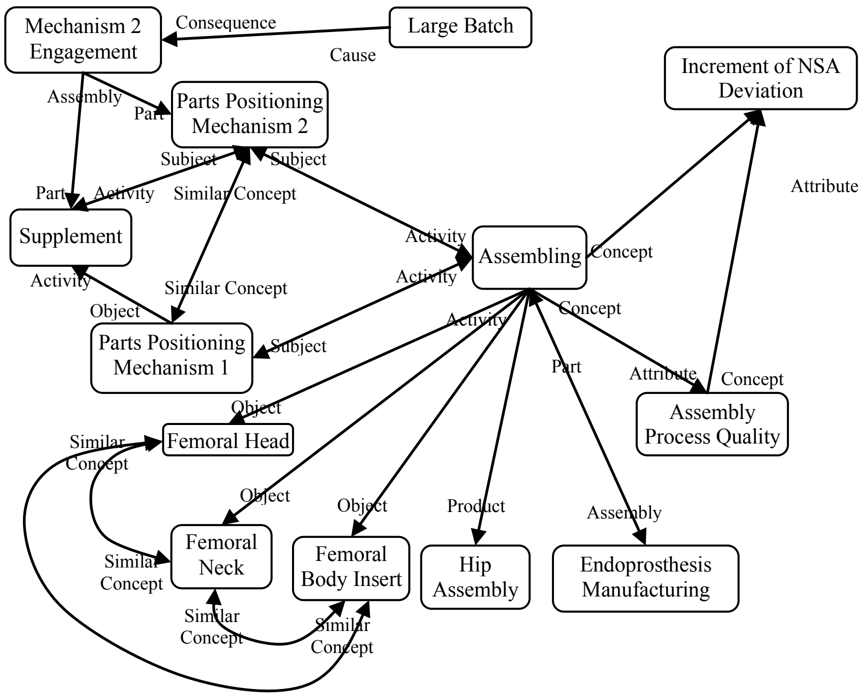

Fig. 12 Associations' plexus (context) that models the context of Endoprosthesis Assembling activity of assembly according to the plexus that models generic Assembling activity

5.3 Applying learned associations in a new situation

By performing the topological analysis of the network, ASM can determine that there are topological analogies between the current situation (described by the input context instance) and the plexuses which are already in the network. Within the same process, ASM determines the degree and the quality of the similarities between certain association plexuses.

In this particular case, ASM recognizes that the context of the activity Scaffold Modeling is topologically similar to the context of the activity Assembling in the current process (at instance level). Following the procedure of upgrading the current context based on the topologically analogous one, ASM initially proposes adding the associations between the concepts Unsuccessful Operation and Large Batch and Small Increment of NSA Deviation, which are already introduced (embedded) in the network (Fig. 13).

The next step is to categorize the new context with additional concepts as a possible exception. ASM suggests making an association between the context which describes the activity instance and the concept Exception. In the process of further upgrade, ASM proposes making a connection between the concept Exception and the context which describes the activity model with the concept Alternate. In the end, the UDF Based Scaffold Modeling and the sequential decomposition of that activity are offered as a way of solving the problem (Fig. 14). These conclusions refer to the context which describes the activity model, and, therefore, they should be applied to all instances of that process. At that moment, the associations with the concept UDF Based Scaffold Design and the concept Sequential Job Decomposition are offered. The administrator will refuse the former one because that connection is not applicable in this context, but he should accept the latter one and consider what this decomposition might refer to in a particular case.

The process of context upgrade is shown in Figs. 13 and 14.

ASM is currently connected with the MD system so as to provide recommendations for solving the problem. These recommendations are then implemented by the rules in the expert systems. These rules are used to change the process in accordance with the recommendations by ASM. If ASM connects a situation with the term Exception and if the concept Alternate also appears, the MD system will send a signal to the expert system that a change in the process is required. The change will be performed by calling an appropriate rule.

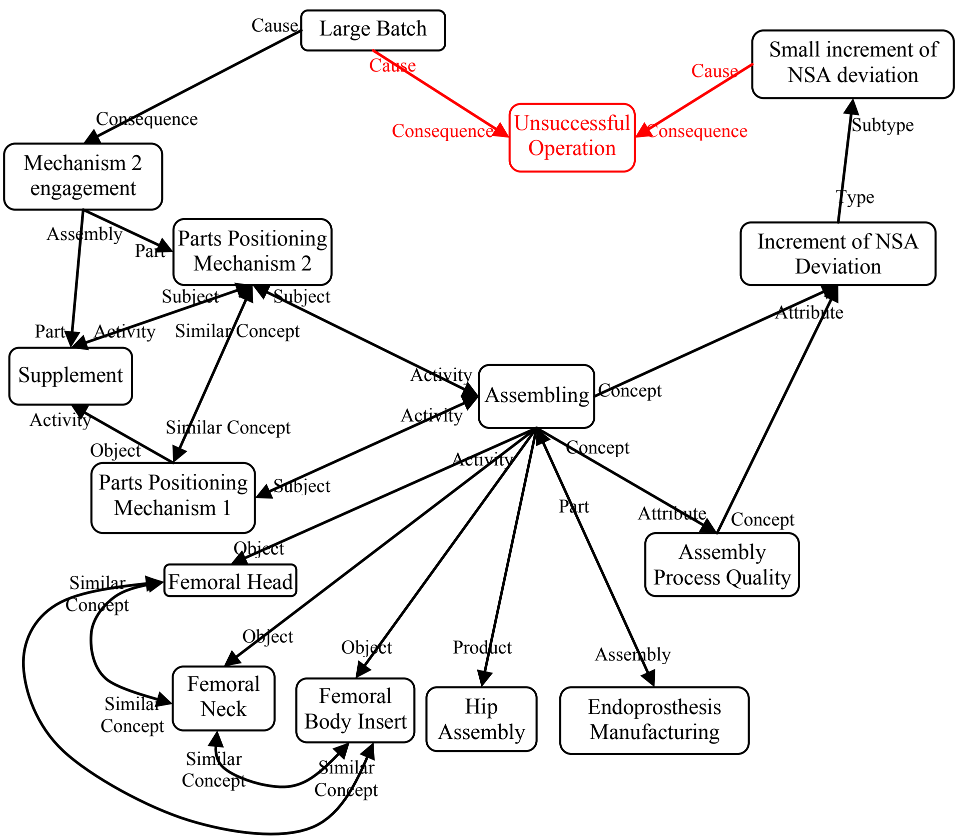

Fig. 13 The context of the activity Assembling with added associations based on the similarity with the context of the activity Scaffold modeling (marked in red)

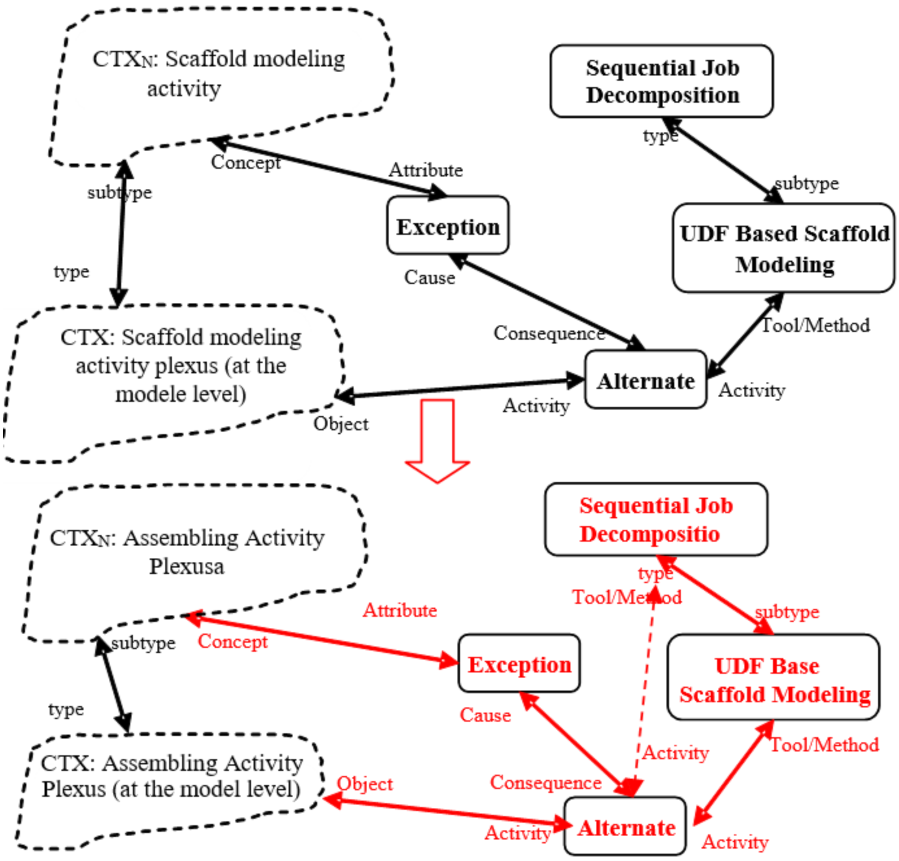

Fig. 14 The upgrade of the piece of network related to the Assembling activity instance which is featured as an Exception according to the partially analogous model of Scaffold Modeling activity instance that is featured as an Exception also

6. Discussion

Our research results related to the application of ASM for detecting and handling exceptions show that ASM brings significant improvements in this domain. In this work, we have shown how a new process (prosthesis implanting) can benefit from the knowledge collected in a difference process (designing a scaffold), where these two processes are not closely connected.

The processes used here as examples for managing exceptions come from close fields (mechanical engineering), but the application of ASM is not limited to such situations. ASM can draw conclusions even in completely different contexts. For example, the conclusion that it is necessary to decompose work could also be drawn from the analogy with construction engineering in the case where a building could not be finished on time, so it was required to split the work.

In our previous work [4], we used the rules of the expert system to represent the process knowledge. The drawback of that approach is that for solving a problem it is necessary to define the rules (knowledge) specifically related to that problem. Collected knowledge can be represented in other ways. Nowadays, ontologies are widely used for this purpose. Instead of relying on ontologies, we have decided to use ASM as a mechanism for knowledge representation and reasoning.

The advantage of ASM over ontologies derives from the fact that ASM imitates human way of thinking, i.e. it is capable of drawing conclusions even on the basis of incomplete information. ASM is thus able to draw conclusions in a new area, on the basis of analogy with an area that is not directly connected to the first one.

Since ASM bases its reasoning on the knowledge that it has previously built into the network, there is a risk of the so-called indoctrination. Negative indoctrination of ASM, which results in the production of incorrect conclusions, can occur in two cases. In the first, the user/teacher can transfer their misconceptions (ignorance) by incorporating their knowledge about a certain domain. In the second case, when the ASM is "taught" about a certain domain by several users/teachers, there is a danger that the semantic content will be inconsistent. This second case is particularly interesting because the growth of knowledge and the semantic network can be significantly accelerated by providing access to ASM *via* the web.

So, the ASM concept cannot guarantee that the expert/teacher did his job rightly just as no one can guarantee to have got completely correct inferences from any kind of an AI method. Unique and completely correct inference is possible just in the case of strictly defined corpus of knowledge, like in formal logic or mathematics. However, for a great majority of situations in the real world, this is not possible.

In the MD system, ASM can be used in two complementary ways: for exception detection and exception handling. Exception detection is explained in a greater detail in [5], while in this work, we made a step further and used ASM for solving challenging situations. ASM in this case manages to recognize a situation as an exception and offers advice based on the analogy with the knowledge previously incorporated into the ASM model.

In addition to offering an intelligent advice, the system which handles exceptions should enable the realization of the offered solution. Sometimes it is possible to do that without human participation, and sometimes it is not. In the example we have presented, the human is left to try to modify the process on the basis of the ASM's recommendations.

The propagation of changes suggested by ASM is currently done *via* expert rules developed previously [4]. These rules enable the process update that can be applied to new or already defined process instances, or a combination of the two. The changes are implemented *via* the Java methods invoked from the action parts of these rules.

7. Conclusion and Future Work

Exception handling is one of the problems that are not solved adequately in the existing business process management systems. The process of solving this problem can be divided into two stages. The most important step is that the system detects an exception in the first stage in order to be able to solve the exception in the second one.

In this paper, we described the MD system that uses ASM and expert rules to handle exceptions. The expert rules are used as a mechanism for handling exceptions that can be predicted in advance. When an exception is detected, a sequence of rules is initiated that modify the process according to the new situation. For detection and handling of exceptions that cannot be predicted in advance, we used ASM. ASM makes conclusions based on the analogy between current and some of the previous situations. For ASM to be able to make conclusions, the business process and all activities involved in it are represented as a semantic network. When a new concept and association are added to the network, a mechanism is triggered to find if that new situation is an exception and if so, to potentially propose a solution based on the similarity with some previous situation. The solution is then forwarded to the system *via* the expert rules that adapt the process.

In our previous work, we began to use ASM for handling of the exceptions. At first, we only used it for detection of the exceptions. One of the reasons for the relatively limited use of ASM at that moment (a few years ago) is the fact that algorithms for recognition of the topological analogies were not fully developed at the time. In the meantime, we have significantly improved these algorithms, and we wanted to show how they can be put to the best use.

In this paper we took a step forward and also used ASM for solving problems. The examples we used for presenting the capabilities of drawing conclusions come from similar processes, but the parts of the processes which are used in analogies are very distant. The situation which is used for ASM's learning is from the area of designing and modeling while the situation in which the collected knowledge is used is from the area of manufacturing. The examples could have been from the processes which are used in a completely different area but we rather wanted to describe a situation which is likely to happen in the same company.

Currently, the ASM's conclusions may only be applied by using the rules from the expert system, but our plan for the future is to enable ASM to independently apply its proposals. Concept bodies may include formalized knowledge structures. This is so-called firmly structured knowledge contained in unambiguous mathematical and logical formalisms transformed into procedural or object-oriented programs. These programs can be called for execution when appropriate.

When it comes to the development of ASM, we plan to further develop procedures for creation of heuristics and knowledge crystallization. We also plan to develop structural elements for semantic categorization of events, i.e. the contexts which come one after another in a certain timeline. ASM could thus be used in the systems where the time dimension has a semantic value.

In addition to the use of ASM for adapting processes to new circumstances, we plan to enable the use of ASM as a support tool in the Adaptive Case Management systems in further work. Initially the system will function in a manner which will let the user define the steps that the process should include; but with the spreading of its knowledge base, ASM will increasingly become able to advise the user about further doings.

Acknowledgements: *This research was financially supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia (Contract No. 451-03-9/2021-14/200109).*

REFERENCES

1. Kir, H., Erdogan, N., 2021, *A knowledge-intensive adaptive business process management framework*, Information Systems., 95, 101639.
2. Sadiq, S., Orlowska, M., 2000, *On capturing exceptions in workflow process models,* In: Abramowicz, W., Orlowska, M. (Eds.) BIS 2000, pp. 3–19.
3. Casati, F., Ceri, S., Paraboschi, S., Pozzi, G., 1999, *Specification and Implementation of Exceptions in Workflow Management Systems*, ACM Transactions on Database Systems 3, pp. 405–451.
4. Mišić, D., Domazet, D., Trajanović, M., Manić, M., Zdravković, M., 2010, *Concept of the exception handling system for manufacturing business processes*, Computer Science and Information Systems, 7(3), pp. 489-509.
5. Mišić, D., Stojković, M., Domazet, D., Trajanović, M., Manić, M., Trifunović, M., 2010, *Exception detection in business process management systems*, JSIR, 69(03), pp. 1038-1042.
6. Dijkman, R., Turetken, O., van IJzendoorn, G.R., de Vries, M., 2019, *Business processes exceptions in relation to operational performance*, Business Process Management Journal, 25(5), pp. 908-922.
7. Ariouat, H., Andonoff, E., Hanachi, C., 2016, *Do Process-based Systems Support Emergent, Collaborative and Flexible Processes? Comparative Analysis of Current Systems,* Procedia Computer Science, Elsevier, 96(C), pp. 511-520.
8. Allen, D., Chapman, A., Blaustein, B., Mak, L., 2015, *What do we do now? Workflows for an unpredictable world,* Future Generation Computer Systems, 42, pp. 1–10.
9. Richter, M., Weber, R., 2013, *Case-Based Reasoning,* Springer, Berlin Heidelberg.
10. Minor, M., Bergmann, R., Görg, S., 2014, *Case based adaptation of workflows*, Information Systems, 40, pp. 142-152.
11. Zeyen, C., Müller, G., Bergmann, R., 2018, *A conversational approach to process-oriented case-based reasoning*, IJCAI'18: Proceedings of the 27th International Joint Conference on Artificial Intelligence, pp. 5404 – 5408.
12. Aha D.W., Breslow, L.A., Munoz-Avila, H., 2001, *Conversational case-based reasoning,* Applied Intelligence, 14, pp. 9-32.
13. Reichert, M., Dadam, P., 2009, *Enabling adaptive process-aware information systems with adept2*, in: Cardoso, J., van der Aalst W., (Eds.), *Handbook of research on business process modeling, information science reference*, New York, pp. 173-203.
14. Laznik, J, Juric, M., 2013, *Context aware exception handling in business process execution language,* Information and Software Technology, 55(10), pp. 1751-1766.
15. Yao, W., Kumar, A., 2013, *CONFlexFlow: Integrating flexible clinical pathways into clinical decision support systems using context and rules*, Decision Support Systems, 55(2), pp. 499-515.
16. Dang, J., Hedayati, A., Hampel, K., Toklu, C., 2008, *An ontological knowledge framework for adaptive medical workflow*, Journal of Biomedical Informatics 41, pp. 829-836.
17. Marrella, A., Mecella, M., 2018, *Cognitive Business Process Management for Adaptive Cyber-Physical Processes,* in: Teniente, E., Weidlich, M., (Eds), *Business Process Management Workshops*, Springer: Berlin/Heidelberg, Germany, pp. 429–439.
18. Stojkovic, M., Trifunovic, M., Misic, D., Manic, M., 2015, *Towards Analogy-Based Reasoning in Semantic Network*, Computer Science and Information Systems, 12(3), pp. 979-1008.
19. Trifunovic, M., Stojkovic, M., Misic, D., Trajanovic, M., Manic, M., 2014, *Recognizing topological analogy in semantic network,* International Journal on Artificial Intelligence Tools, 24(03), pp. 1550006-1 – 1550006-25.
20. Trifunović M., Stojković M., Trajanović M., Mišić D., Manić M., 2013, *Interpreting the meaning of geometric features based on the similarities between associations of semantic network,* Facta universitatis, Series: Mechanical Engineering, 11(2), pp. 181-192.
21. https://sourceforge.net/projects/sharkwf/ (last access: 15.12.2021)