Management and Production Engineering Review
Volume 5 • Number 1 • March 2014 • pp. 51–64
DOI: 10.2478/mper-2014-0007

SERVICE CONFIGURATION IN SOA-BASED ENTERPRISE REPRESENTATION USING ROLE KNOWLEDGE

Kurt Sandkuhl[1,2], Ulf Seigerroth[1], Alexander Smirnov[3], Tatiana Levashova[3], Nikolay Shilov[3]

[1] *School of Engineering, Jönköping University*
[2] *Institute of Computer Science, Rostock University*
[3] *St. Petersburg Institute of Informatics and Automation of the Russian Academy of Sciences*

Corresponding author:
Nikolay Shilov
St. Petersburg Institute of Informatics and Automation of the Russian Academy of Sciences
39, 14 Line, 199178, St. Petersburg, Russia
phone: (+7 812) 328-80-71
e-mail: nick@iias.spb.su

Received: 11 November 2013
Accepted: 12 February 2014

Abstract

This work investigates the use of enterprise models for service configuration with a focus on knowledge related to organizational roles. Starting from an industrial case, it presents an approach to extract a role's information demand from an enterprise model and to use this as initial configuration for agent-based services. In addition to the meta-model included in the enterprise modeling language, the use of a common ontology is proposed which captures both, the perspective of information demand and services. The architecture of the agent-based services is following the idea to characterize all actors by their roles and to represent them by sets of services. This approach facilitates self-organization in the service level. The main contributions of the paper are (1) to show that a role's information demand is relevant for service configuration, (2) to present a way of extracting information demand from enterprise models, and (3) to extend an approach for SOA-based enterprise representation with information demand.

Keywords

enterprise model, role, information demand, service-oriented architecture, service configuration.

Introduction

Enterprise modeling, enterprise architecture, and business process management are three areas that for a long time have been part of a tradition where the mission is to improve business practice and management [1]. In general terms, enterprise modeling is addressing the systematic analysis and modeling of processes, organization structures, products structures, IT-systems or any other perspective relevant for a given modeling purpose [2]. The knowledge about enterprises captured in enterprise models usually is starting point for organizational change or improvement activities, or is used in information system development. In this context, the use of process models for service discovery, service composition or service verification has received much attention. In order to achieve this, the information needs to be accurate, pertinent and readily available. A prerequisite for providing relevant information is the understanding of the information demand different actors in enterprises have as well as the information flow aiming at meeting this information demand [3].

This paper investigates a different perspective of this area: the use of knowledge about organizational roles captured in enterprise models. Work in information logistics showed that information demand of an organizational actor depends on his tasks and responsibilities [4], i.e. on the organizational role. At the same time, recent studies showed that information overload more and more is perceived as a problem in enterprises. An example is a study among Swedish enterprises where 2 out of 3 top or mid-level managers perceive an information overload due to "far too much information" (37%) or "too much information" (29%) [5]. In such a situation, using information about a role's organizational working context for tailoring information supply and other supporting services is supposed to create substantial benefits. Based on a motivating case from industry, we present a way for extracting a role's information demand and for using this as initial configuration for agent-based services and the implementation of a service-based architecture. The main contributions of this paper are (1) to show that a role's information demand is relevant for service configuration, (2) to present a way of extracting information demand from enterprise models, and (3) to extend our approach for SOA-based enterprise representation with information demand.

The paper is structured as follows: Sec. 2 introduces a motivating case from industry. Section 3 describes information about roles represented in enterprise models and Sec. 4 discusses the extraction of this information from such models. Section 5 presents the SOA-based enterprise representation extended by information demand. Section 6 extends this approach towards self-organization of web services. Section 7 discusses related work and experiences. Section 8 concludes the work.

Industrial case

In this section the industrial case motivating the work in this paper is presented. The case is selected from the research and development project InfoFlow-2, which included 6 industrial and academic partners. Within InfoFlow-2[1], enterprise modeling was performed in a number of cases with focus on capturing information flows and information demands in different industry domains. This section will briefly discuss one of these cases in order to illustrate modeling of knowledge about organizational roles in their enterprise context and the information demand related to the organizational roles in this context. The industrial case selected is taken from manufacturing industries and based on one of the industrial partners of the InfoFlow-2 project. This partner is a sub-supplier to different first-tier suppliers in automotive and telecommunication industries. This InfoFlow-2 partner performs various surface treatment services of different metal components. Surface treatment in this context includes different technical or decorative coatings that are applied to the components in order to achieve certain functionality or appearance.

The industrial case was focusing on engineering change management (ECM) in the production process. The case partner is a subcontractor to many different OEMs[2] in the automotive industry. The main challenge for the case partner is to handle the continuous flow of incoming changes of specifications for the products produced for the OEM. The situation for the sub-supplier is more like continuous change than stable business processes. Not implementing changes in time would lead to defective products and economic consequences for the sub-supplier. This situation with constant change of specifications was our overall focus for the information demand analysis. But in order to really get into details we also modeled a specific part of the ECM process (from change request to production planning). This enabled us to address the information demand for different roles as they are performing specific assignments (activities). The information demand was performed using a specific method developed for this purpose [3].

The actual modeling was divided into two main activities: 1) interviews, and 2) facilitated information demand modeling sessions. For these two activities we had selected the roles that significantly were involved in the process section (from change request to production planning). These roles were head of quality, sales representative, technical support/technical in-house sales, and production planner. In addition to this, we also had two researchers involved in the modeling activity. The interviews with the selected roles were performed in a semi-structured manner during two hours where one of the researchers was guiding the interview and the other researcher took down notes in a logbook. The main purpose of the interviews was to reach an understanding of the current situation that was "good enough", and to set the stage in terms of deciding about the focus for the following modeling session. The major purpose of the modeling session was to describe the information demand for different roles based on their assignments and responsibilities in the process. The modeling session was performed in a participative way where the representatives from the industrial partner were actively involved in the modeling activity. The result from the session was for instance used by the head of quality to illustrate and share knowledge among the stakeholders about certain aspects in the change management process. The models that were developed during the information demand modeling session served as an instrument to evaluate the current practice and to suggest improvements related to information flow.

[1] www.jth.hj.se/infoFLOW.

[2] OEM = Original Equipment Manufacturer.

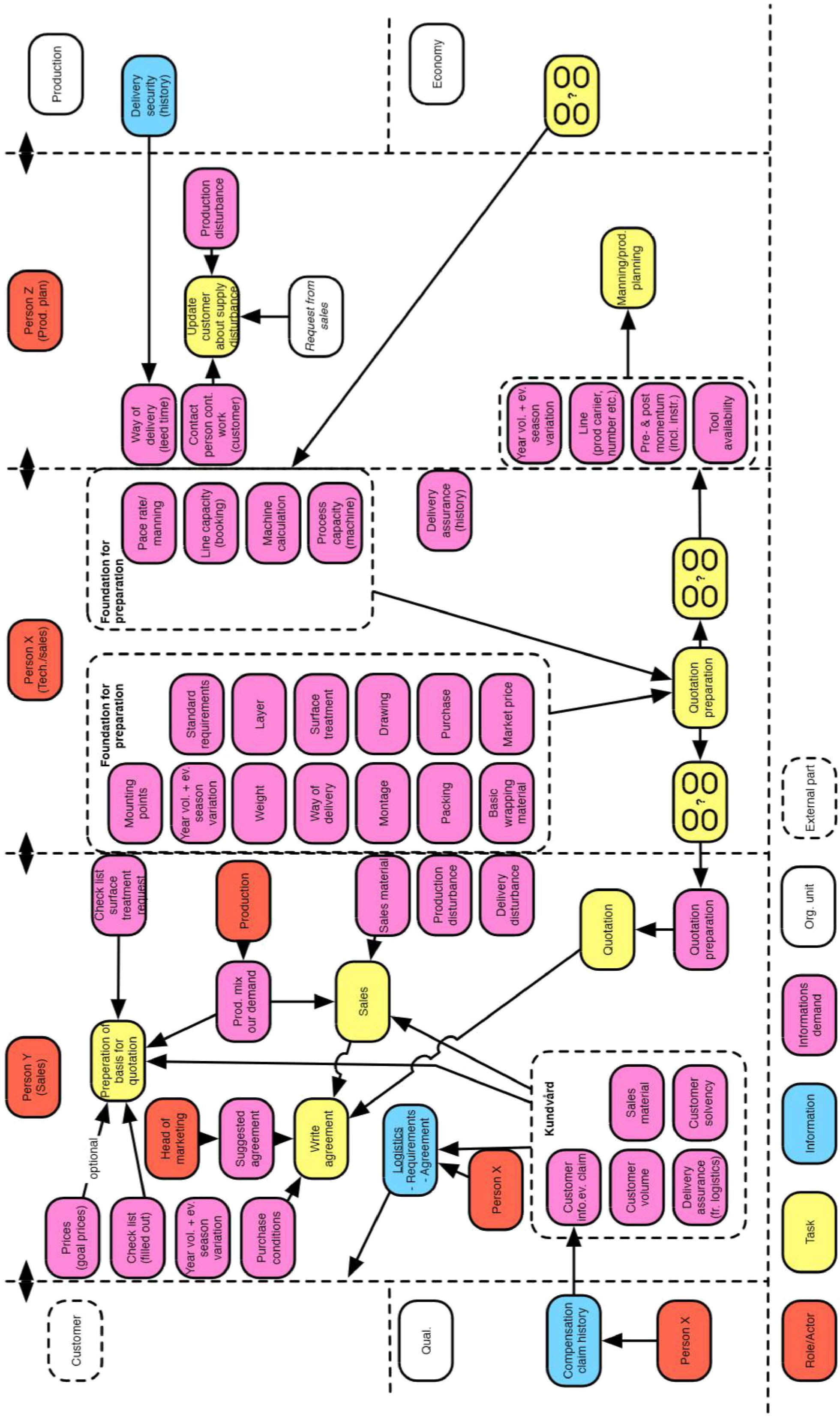

Fig. 1. Excerpt of the model developed in the industrial case.

Figure 1 is part of the model developed in the industrial case focusing on the sales process (from change request to production planning). The model shows the organizational roles involved from different departments: sales, technical planning, production planning and customer care. The model also shows how these roles are interacting in terms of information flow and information demand during preparation of a quotation for a change request. The principle in the produced models and the notation rules is that for each role and for all the assignments or tasks (process activities) the model should identify and illustrate the information required to perform a certain task. In this case, we modeled both information that were available and information that was not available. This way of modeling also visualized where the missing information was supposed to come from. We also identified and modeled the information that was provided by performing a task and in some situations it even became apparent that some information really wasn't needed. In one case it became clear that the receiving role did not understand that he/she actually needed this information to perform a certain task. As a result of this, the notation rules for information demand analysis were refined so that it can cover and express a greater variety of relations in the model.

Information demand in enterprise models

The enterprise model developed in the industrial case contains elements and relationships indicating the information demand of the roles involved. The understanding and definition of the term information demand used in this paper is based on empirical work performed during 2005–2007, which also contributed a deeper understanding of how information is used with regards to work-related tasks. Information demand will be used throughout this paper with the following meaning: "*Information Demand is the constantly changing need for relevant, current, accurate, reliable, and integrated information to support (business) activities, whenever and where ever it is needed*" [6].

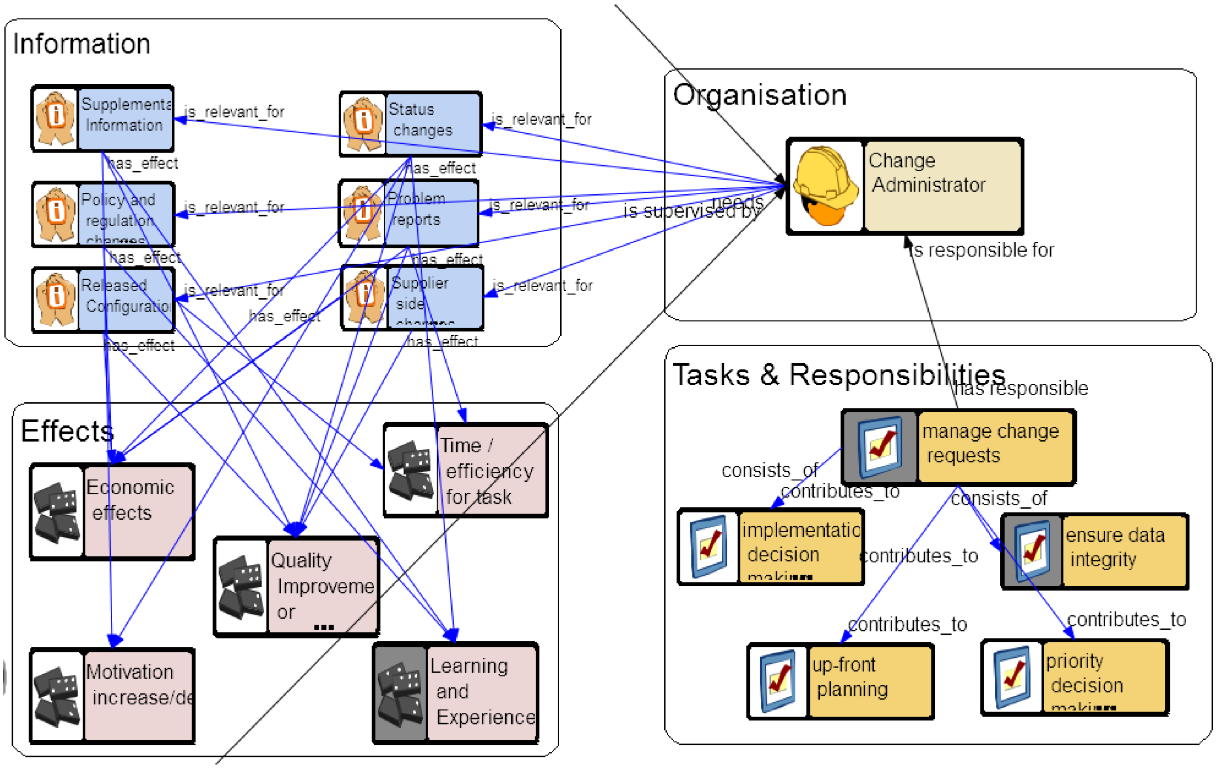

Fig. 2. Information demand context for the role change administrator (adopted from [7]).

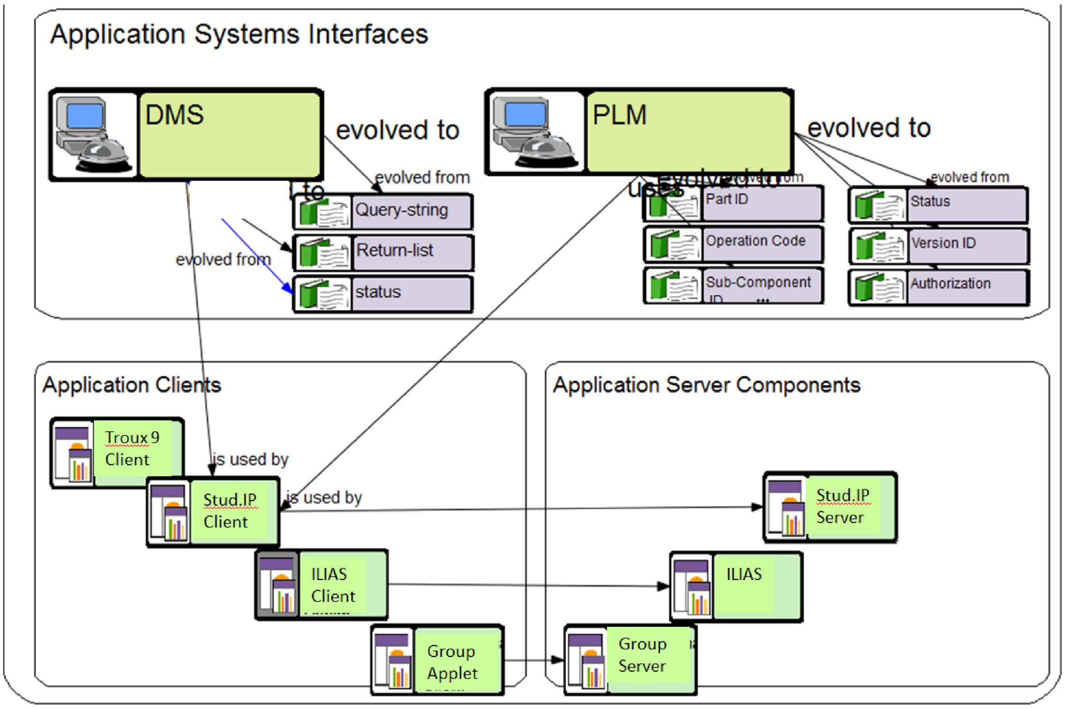

Fig. 3. Infrastructure sub-model with web-service interfaces.

Furthermore, the above mentioned empirical work confirmed the conjecture that information demand of a person is based on the roles and tasks this person has: "*Information demand depends on the role and tasks an entity has within a larger organization. If the role and/or the tasks change, so too will the demand*" [6]. This role-centric perspective with task and responsibilities as primary characteristics has been the guiding criteria for extracting those parts of the enterprise model relevant for a specific role. Figure 2 shows such an information demand centered sub-model for the role of a change administrator. In the continuation of the article this type of demand-focused and role-specific model will be denoted as information demand context.

The upper right container in Fig. 2 shows the role "change administrator". Linked to this role, the sub-model includes

- tasks and responsibilities (lower right container) of the role, like "ensure data integrity" for change documents,
- relevant information for the tasks (upper left container), like "status changes" of change projects, and
- effects of not receiving the information (lower left container), like economic effects.

Linked to the information demand context, we find in the enterprise model the IT-resources needed for performing specific tasks or process steps. These resources are modeled in the infrastructure sub-model of the enterprise model with their web-service interfaces. Figure 3 shows such an infrastructure sub-model, which among other web-service interfaces contains a PLM (Product lifecycle management) and a DMS system (document management system).

With the link from a role's information demand to the IT-resources and web-services used by the role we do not only find important information for a role's profile, but also an initial configuration, which services have to be considered relevant from the role's perspective.

Extracting information demand for a role

As a precondition for using role knowledge captured in enterprise models for service configuration, the relevant portions of information first have to be extracted from enterprise models. This information is usually represented in enterprise modeling languages such as EEML, UEML or MEAF. Information extraction deals with the problem of extracting information from documents for a given question [8]. Information extraction (IE) approaches can be divided into those "that require human intervention at run time, and those that require little or no intervention. Full automation is not always necessary in order to produce a useful tool, and may be undesirable in tasks requiring human judgment" [9]. Appelt and Israel [10] suggest that there are two basic approaches for information extraction: the knowledge engineering approach and the automatic training approach.

The main characteristic of knowledge engineering approaches is that a 'knowledge engineer' develops the grammars used by a module in the IE system. The knowledge engineer is someone familiar with the IE system and the formalism for defining rules for that system. The task is to write rules (for the IE system component) that mark or extract the sought-after information. The skill of the knowledge engineer is a crucial factor in the level of performance that will be achieved by the overall system. Regarding the advantages of this approach, one could mention that "with skill and experience, good performing systems are not conceptually hard to develop" [10] and "the best performing systems have been hand crafted" [10]. But there are also disadvantages of using this way of building IE systems. Firstly, it is a very time-consuming development process (it requires a quite demanding test-and-debug cycle); secondly, changes to specifications can be hard to accommodate and thirdly the required expertise may not be available.

In automatic training approaches, it is not mandatory "to have someone on hand with detailed knowledge of how the IE system works, or how to write rules for it. It is necessary only to have someone who knows enough about the domain and the task". According to [10], this person's role is "to take a corpus of texts, and annotate the texts appropriately for the information being extracted". The result of the annotation of a corpus is called 'training corpus'. Once a suitable training corpus has been annotated, a training algorithm is deduced from it, resulting in information that a system can apply in analyzing novel texts. Thus system expertise is not required for customization, because if someone familiar with the domain is available to annotate texts, IE systems can be built for a specific domain. Another advantage of this approach is that 'data driven' rule acquisition ensures full coverage of examples. This originates from the fact that the user, first, annotates the text and by doing so, gives all the examples of the sought-after information to the system. The disadvantages of the automatic training approach is that large volume of training data may be required and it may happen that changes in specifications require re-annotation of large quantities of training data.

In order to extract role knowledge from enterprise models, we can use experiences from previous work aiming at applying such models for team formation. The information extraction software component built for this purpose is described in [11] and can easily be adapted for this purpose.

When developing this software component, the purpose was mainly to show feasibility of enterprise model usage for information extraction. The development was guided by one primary use case, which assumed that an existing enterprise model would be the input for the system to be developed. Not the complete model is used for information searching, but the user specifies the area of interest in the model by selecting an entity type to be considered. By doing this, the user selects a sub-model that consists of the selected entity type and all entity types connected by relations to this entity type. A sub-model could for example be the entity type "role" plus the entity types such as "task" connected via a relation "has_task" to "role".

When designing the software component, a modular architecture was chosen because it provides great flexibility both at the implementation and during the maintenance phase. Figure 4 summarizes the different components of this architecture.

The software uses an existing Enterprise Model (EM) to find possible candidates for a given role. First of all, the user selects an Enterprise Model and a role. The model and the role selected are given as input for the first module, which is called EM Analyzer. This module uses the given role to extract information attached to the role from the model file. Once the competences are extracted, they are used as input for the second module, the Profile Constructor. The function of this module is to first identify services in the service-based enterprise representation (see next section) to be configured with role-based information and then to provide the information extracted to these services. Furthermore, the User Interface (UI) has been developed to support all the interactions between the user and the system. The purpose is to visualize what information has been extracted and which services could be configured.

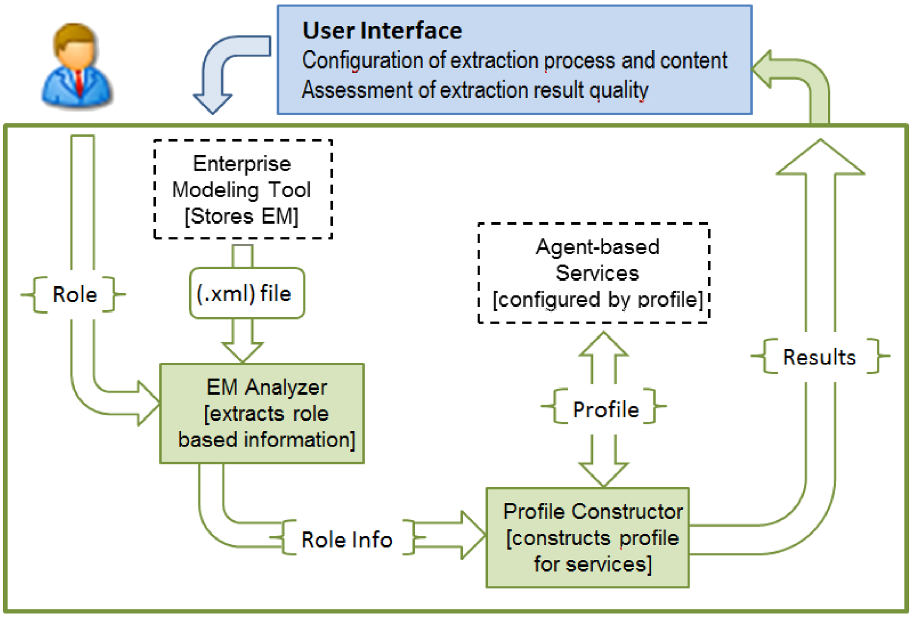
User Interface
Configuration of extraction process and content
Assessment of extraction result quality
Enterprise Modeling Tool [Stores EM]
Role
(.xml) file
EM Analyzer [extracts role based information]
Agent-based Services [configured by profile]
Profile
Results
Role Info
Profile Constructor [constructs profile for services]

Fig. 4. Overall architecture of the developed IT solution (adapted from [11]).

SOA-based enterprise representation

The SOA-based representation assumes characterizing all actors by their roles and describing them via profiles (Fig. 5). The profiles are associated with agent-based services that negotiate to take into account explicit and tacit preferences of the roles. This makes it possible to replace the information sharing between the actors with that of distributed services.

The representation integrates efficient sharing of information with the service-oriented architecture taking into account the dynamic nature of the enterprise environment. For this purpose, the models proposed are updated in accordance to the current situation. An ontological model is used to solve the problem of service heterogeneity. The ontology is usually built in a semi-automatic way based on existing databases and electronic documents. This model facilitates interoperability between heterogeneous information services due to provision of their common semantics and terminology. Application of the context model makes it possible to reduce the amount of information to be processed. The abstract context essentially describes the information demand to be provided or received by the actors. This model enables management of information relevant for the current situation. The access to the services, information acquisition, transfer, and processing (including integration) are performed via usage of the technology of Web-services.

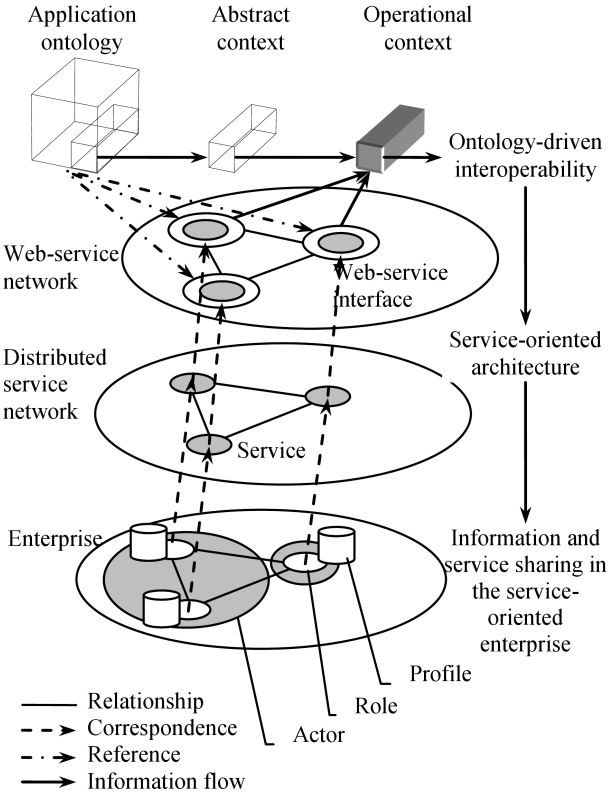

Fig. 5. SOA-based enterprise representation.

In our industrial case, the implementation of information sharing between the organizational roles via distributed services could enable automated service composition based on the information demand (the self-organisation protocol is presented in the next section). The enterprise model contains both the information to be shared and the services connected to this information. The role of a change administrator, to take an example, needs information about status changes, which would be provided by a PLM system documenting the product information and ongoing change projects. This abstract set-up taken from the enterprise model could be configured to an operational one by taking the information about the individual having the role of change administrator, since the actual employees are modeled in the organizational sub-model.

Self-organization of Web-services

The analysis of literature related to organizational behavior and team management showed that the most efficient teams are self-organizing teams working in the organizational context.

The approach is based on the following principles: self-management and responsibility, decentralization, as well as network organization (without any social hierarchy of command and control within a level), and co-operation within one level.

Self-organization is considered as a threefold process of (i) cognition (where subjective context- and role-dependent knowledge is produced), (ii) communication (where system-specific objectification or subjectification of knowledge takes place), and (iii) synergetic co-operation (where objectified, emergent knowledge is produced). The individually acquired context- and role-dependent knowledge (subjective knowledge) can be put to use efficiently by entering a social co-ordination and co-operation process. The objective knowledge is stored in structures and enables time-space distanciation of social relationships. As a result, social environments persist much longer than an average individual (actor) contributes to it. Nevertheless, they are flexible and dynamic in character as well as self-contextualized.

In order to achieve the dynamics and self-contextualization of the self-organizing system, its components (actors) have to be creative, knowledgeable, active, and social. The individuals that are parts of a system permanently change their joint environment in accordance with their roles what results in a synergetic collaboration and leads to achieving a certain level of collective intelligence. This is also supported by the fact that individual behavior is partially determined by the social environment the individuals are contributing to (called "norms").

The architecture [12], supporting a decision situation, distinguishes two kinds of Web-services: *core* Web-services and *operational* Web-Services.

The core Web-services are intended for creating, storing, maintaining the abstract context (abstract context service), and registering the Web-services (registration service).

The operational Web-services are intended for representing properties and functions of the actors, self-organization of a Web-services' community, instantiating the abstract context, planning and taking the joint actions. Main types of Web-services comprising the set of operational Web-services are 1) Information Resource and Problem-Solving Resource (IPSR) services and 2) acting resource services.

To make the Web-services active components capable of self-organizing an agent-based service model [13] is used. Agents negotiate the services' needs and possibilities and "activate" Web-services when required.

The starting point for self-organization of Web-services is the abstract context. This context is created based on the information demand.

As soon as the abstract context has been created, the operational Web-services start organizing the community of Web-services. It is organized following three scenarios running one after another (Fig. 6). In these scenarios, the following information about the resources is used: competences of the resources, roles the resources fulfill, the resources' availabilities, and the resources' weights. This information is represented by their profiles. In the following, "information about resources represented by Web-services" will be referred to as "information about Web-services", e.g. web-service availability means the same as availability of a resource this Web-service represents.

The first scenario "Scenario 1" describes the Web-services' interactions in order to select from the set of contextual Web-services WS_c a set of Web-services WS_f that implement functions required to instantiate the classes and to solve the problems specified in the abstract context, and that fulfill the specified roles.

The second scenario "Scenario 2" describes the interactions of the Web-services comprising the set WS_f in order to select from this set a set of Web-services WS_a that would be available over a predicted time interval $[t_0, T]$ (the interval of adequacy of the decision situation model), with t_0 – the time instant when the Web-services start initiating the abstract context, T – the time instant when the Web-services come to the decision about starting the joint actions. The result of the second scenario is a set of Web-services (WS_a) that are partly or fully available during the predicted interval.

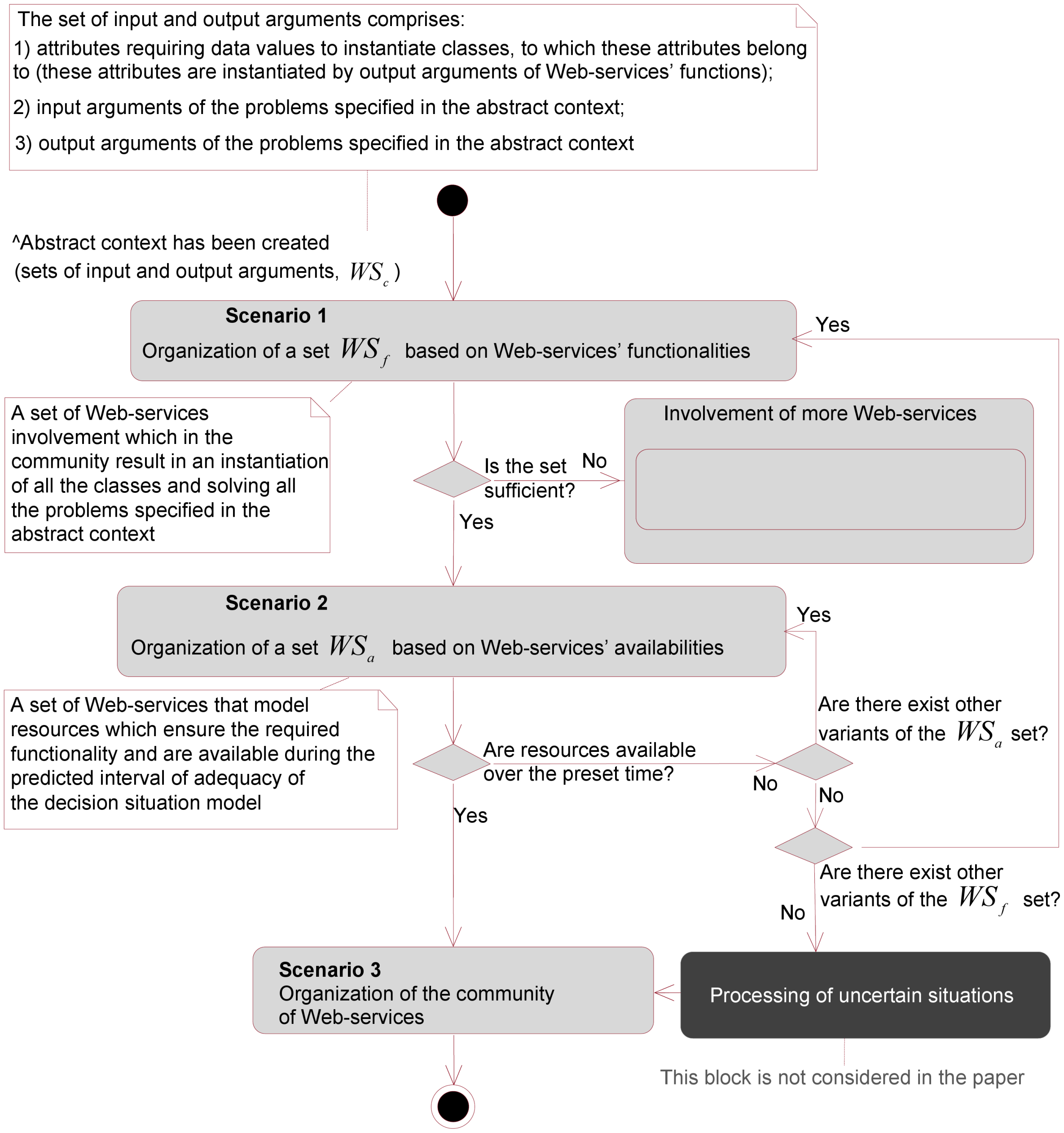
The set of input and output arguments comprises:
1) attributes requiring data values to instantiate classes, to which these attributes belong to (these attributes are instantiated by output arguments of Web-services' functions);
2) input arguments of the problems specified in the abstract context;
3) output arguments of the problems specified in the abstract context
^Abstract context has been created (sets of input and output arguments, WS_c)
Scenario 1
Organization of a set WS_f based on Web-services' functionalities
Yes
A set of Web-services involvement which in the community result in an instantiation of all the classes and solving all the problems specified in the abstract context
Is the set sufficient?
No
Yes
Involvement of more Web-services
Scenario 2
Organization of a set WS_a based on Web-services' availabilities
Yes
A set of Web-services that model resources which ensure the required functionality and are available during the predicted interval of adequacy of the decision situation model
Are resources available over the preset time?
Yes
No
Are there exist other variants of the WS_a set?
No
Are there exist other variants of the WS_f set?
No
Scenario 3
Organization of the community of Web-services
Processing of uncertain situations
This block is not considered in the paper

Fig. 6. Scenario of organization of Web-services community.

The third scenario "Scenario 3" describes interactions of the Web-services comprising the set WS_a for the purpose of efficient organizing the Web-services community. This community is organized based on the following principles.

Principle of maximum functionality. For the Web-services that implement several functions or fulfil several roles, their involvement in the Web-services community of a specific Web-service implementing several functions or fulfilling several roles is considered more expedient than involvement of several resources implementing the same functions or fulfilling the same roles separately, i.e. $|WF_n \cap FN| \rightarrow \max$ for the n^{th} Web-service, where WF_n – the set of functions the n^{th} Web-service implements, FN – the set of functions specified in the abstract context; and $|WR_m \cap RN| \rightarrow \max$ for the m^{th} Web-service, where WR_m – the set of roles the m^{th} Web-service fulfills, RN – the set of roles specified in the abstract context.

Principle of maximum access interval. If several Web-services exist that are accessible over the interval $[t_0, T]$ at different time intervals $\{\Delta t\}$, then selection of less number of Web-services whose overall access intervals are closest to the interval $[t_0, T]$

is considered to be more efficient. At that, the interval $[t_0, T]$ must be fully covered by the intervals Δt_i^n (Δt_i^n – the i^{th} access interval for the n^{th} resource, $\Delta t_i^n \in \{\Delta t\}$), i.e. $[t_0, T] \subseteq \bigcup_{n=1}^{NR} \bigcup_{i=1}^{nt} \Delta t_i^n$ (nt – number of intervals Δt_i^n over the interval $[t_0, T]$ for the n^{th} resource, NR – number of Web-services in the set WS_a).

Principle of minimum weight. This principle is used to evaluate the efficient selection of alternative Web-services. Alternative Web-services are Web-services implementing the same functions or fulfilling the same roles, but differing in their locations, costs, *etc.* Weight of a Web-service is calculated as $W = \alpha(1 - N) + \beta T_r + \gamma C$, where N – the Web-service's competence / reliability, $N = (0, 1)$; T_r – the average access time to the Web-service relatively to the access time that is maximum among the access times for the alternative Web-services; C – the cost of information acquisition from the Web-service relatively to the cost of information acquisition that is maximum among the costs for the alternative Web-services; α, β, γ – relative importance of the parameters for the particular Web-service ($\alpha + \beta + \gamma = 1$).

Self-organization of the Web-services is sustained by a self-organization protocol. The protocol supports 5 operations. An operation is implemented as a set of requests, replies, and notifications sent/received by the Web-services to execute the operation. The following operations are supported:

Function – selects from the set WS_c a set WS_f of IPSR services based on the services' functionalities;

Role – selects from the set WS_c a set WS_f of acting resource services based on the services' roles;

Availability – selects from the set WS_f a set WS_a of Web-services representing any kinds of resources based on the services' availabilities over a predicted interval $[t_0, T]$;

Weight – selects from a set of the alternative services containing in the set WS_a a set of Web-services based on the services' weights, so that the resulting set of Web-services would not contain the alternative ones;

Activity – supports activities of the Web-services within the community.

Figure 7 illustrates the operation **Function**. During this operation the abstract context service and three IPSR services interact to select Web-services that satisfy the principle of maximum functionality. The figure contains the following operations:

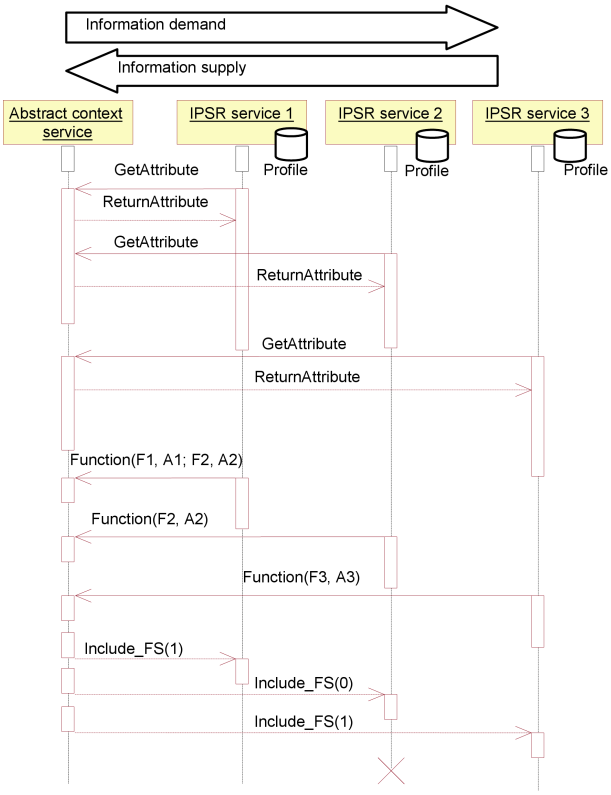

Fig. 7. An example of Web-services messaging when forming a set of IPSR services based on services' functionalities.

GetAttribute is a request of IPSR service to the abstract context service for a list of class attributes occurred in the abstract context;

ReturnAttribute is a reply of the abstract context service to IPSR service in the form of the list of class attributes requiring to be instantiated;

Function is a notification sent by IPSR service to the abstract context service in the form of the list of functions (F1, F2, F3), which results (values of the functions' output arguments) can be used for instantiation of the class attributes (A1, A2, A3) contained in the abstract context;

Include_FS is a reply of the abstract context service notifying an IPSR service about its inclusion in (1) / exclusion from (0) the Web-services community based on the principal of maximum functionality.

As illustrated in the figure, IPSR service 2 has not been included in the Web-services community since to instantiate all the class attributes containing in the abstract context it is sufficient to take the results of the functions implemented in IPSR service 1 and IPSR service 3.

Discussion

In the area of self-organization of Web-services, the research communities currently focus on automated composition of Web-services. Techniques of the automated composition detect the user needs, find out appropriate Web-services in the Internet, compose them in the right order, and execute them [14]. Generally, methods of service composition use techniques of workflow or AI planning [15].

The workflow technique relies upon an abstract model. Such a model describes the user needs in terms of wanted functionality (e.g., by describing information about a set of tasks to be executed). A service composer finds a sequence of services that jointly are capable of providing the requested functionality. Some examples of approaches exploiting the workflow technique can be found in [16–20].

In the AI planning the problem of automatic service composition is formulated as follows. Given the possible actions, which may be executed (a domain theory), the initial state and user goal needed is to find a sequence of Web-services achieving this goal [21]. A lot of research papers investigate the issue of automatic service composition as the AI planning problem, e.g. [22–26].

Lately, a step towards decentralized architectures and distributed algorithms for automatic service composition has been taken [27]. Some of these approaches deal with agent-based services [28–30]. These services are capable of cooperating with the purpose of fulfilling the user needs.

The approach presented in this paper uses an abstract model for representation of the requested functionality. The approach takes a step towards an automated service composition in the sense of the service consumers not having to select appropriate services and link them together. Though, a premise, the approach built upon, is the desired Web-services already exist and are known; there is no need to search them in the open environments. As a way of automated service composition self-organization of Web-services is proposed. To make Web-services active entities an agent-based service model is used. Self-organization of Web-services is supported by a protocol. The protocol allows the Web-services to efficiently organize a collaborative environment.

Two factors which obviously contribute to the quality of service configuration are the content of the service description and the richness of the role's profile description. Quality in this context means how well the services selected for supplying information for a role fit to the individual and organizational needs. We consider detailed competence information as one possibility to improve the role's profile. If the competence needed for having an organizational role is well understood and with all details represented in the enterprise model, it can also be extracted and used for service configuration. Most enterprise modeling languages do not include specific competence-related types in their meta-model. In order to compensate for this deficiency, extensions have been proposed, like in the Unified Enterprise Competence Modelling Language (UECML) [32]. UECML has integrated the concept of competence in the language at three levels; the competence itself, individual competence, and aggregated competence of a group of individuals. UECML is quite unique to explicitly integrate human competence in enterprise models. Integrating UECML constructs into our approach could be one way of improving the quality of competence information in role profiles. In the field of service description, a number of approaches have been proposed for enriching conventional descriptions, like UDDI and WSDL for web services. Selected approaches in this area are Semantic Annotations for WSDL and XML Schema (SAWSDL) [33], ontology-based service annotations [34] or source code annotations [35]. Combining rich competence profiles and rich service descriptions are considered a promising path to improve the quality of the service configuration.

Conclusions

The paper presented a way of extracting a role's information demand context from an enterprise model in order to use this context as initial configuration for agent based services. Furthermore, the extension of this SOA-based approach towards self-organization was presented.

A central part in this work has been to identify and elaborate on services based on a role-centric information demand analysis. The suggestion is that configuration of services should be done so that they meet individual needs of the different organizational roles that individuals can have. To achieve this we have in this paper proposed an approach that starts from the models produced during an information demand analysis. On a conceptual level we see a great potential in this approach since the services will be grounded in and generated from the actual information demand for different roles. There is however, also a challenge in this in terms of the current focus, level of abstraction, the work procedures, and alignment between models in the information demand analysis. In this respect, there is a need to focus the current information demand analysis more towards the dimension of infrastructural support for different assignments. A service should be a purposeful infrastructural support for actions performed by a role. This dimension therefore needs to be elaborated in information demand analysis so that infrastructural support for assignments (actions) is described on a more detailed level. A more detailed level of description could also solve the challenge regarding level of abstraction, which is an open issue in the current version of information demand analysis. In the version used to model in the industrial case it was difficult to reach a refinement level detailed enough to be able to identify services and their detailed characteristics. One way to improve this is to further qualify the relations between roles and their information demand to perform a certain task. In section two we have shown a suggestion to make this improvement in the refinement of the notation rules. Another issue is the work procedures in the current version of the method, which are described to encompass two main activities, interviews and an information demand modeling session. This is adequate as procedure, which aims at instruction about how to proceed. It does however, not give any instructions about how to actually go about the interview and the modeling session in order to capture the information need and, later in the process, the services. The interview for instance, needs to be a semi-structured interview with a number of predefined categories with thematic questions to facilitate and to help to direct the focus in the interview. The modeling sessions then need to be driven by the results from the interviews. This means that the interviews must go through an initial analysis in order to provide useful information about what aspects of the practice that should be the subject of information demand modeling.

With regard to this challenge we have used Figs. 1, 2, and 3 to present three different types of models where a description of information demand is translated into service specifications. In order to ensure that the defined services really capture and are aligned with the initial identified information demand they need to be based on a common ground. In addition to the meta-model included in the modeling language, we suggest a common ontology that captures both the perspective of information demand as well as services. This is also in line with previous work for creating alignment between models for business and IT [31].

Some elements of the proposed approach have been implemented in a project with a global automation equipment manufacturer with more than 300 000 customers in 176 countries. The detailed description of this project is presented in [36, 37]. The automation equipment manufacturer has a large number of configurable products consisting of various components. The roles that have been identified as important in an enterprise when it comes to design and configuration of new products are the product manager and the production engineer. The product manager is responsible for the short-term and mid-term development of a product in terms of translating customer requirements to product features (target setting), guidance and supervision of the design process from customer features to the way of implementing them as functions, control of variability indicated by new and existing product functions, interface to sales and marketing, etc. If changes in the environment occur, the product manager has to investigate options on product level how to act or react. In cooperation with the product manager, the production engineer is responsible for making the product "manufacturable", i.e. adjust product design, material or features to what the machines in the production system can manufacture. Furthermore, the production engineer designs the overall production system by developing an appropriate composition of sub-systems and the flow of resources, materials and products.

The model built enabled automation of a number of processes previously done manually. The main advantages of the developed solution were [36]:

- Automatic creating master data in SAP models;
- Automatic creating data for the configuration models and services;
- Automatic generating an ordering sheet for the print documentation (this ordering sheet was generated earlier with much effort manually);
- Automatic generating a product and service list, which is needed in the complete process implementing new products.

Some parts of the work have been sponsored by grants # 12-07-00298-a, # 12-07-00302-a, # 11-07-00045-a, and # 11-07-00058-a of the Russian Foundation for Basic Research, project # 213 of the research program "Intelligent information technologies, mathematical modelling, system analysis and automation" of the Russian Academy of Sciences, and project 2.2 "Methodology development for building group information and recommendation systems" of the basic research program "Intelligent information technologies, system analysis and automation" of the Nanotechnology and Information technology Department of the Russian Academy of Sciences.

References

[1] Harmon P., *The scope and evolution of business process management*, in Handbook on Business Process Management 1: Introduction, Methods, and Information Systems, J. vom Brocke and M. Rosemann, Eds., Berlin and. Heidelberg, Germany: Springer, pp. 83–106, 2010.

[2] Vernadat F.B., *Enterprise modelling and integration*, Chapman & Hall, 1996.

[3] Lundqvist M., Sandkuhl K., Seigerroth U., *Modelling information demand in an enterprise context: method, notation, and lessons learned*, in Int. J. of Information System Modeling and Design (IJSMD), 2, 3, 75–95, 2011.

[4] Lundqvist M., *Information demand and use: improving information flow within small-scale business contexts*, Licentiate Thesis, Dept of Computer and Information Science, Linköping University, Linköping, Sweden, 2007.

[5] Öhgren A., Sandkuhl K., *Information overload in industrial enterprises – results of an empirical investigation*, in Proceedings of ECIME 2008, London, UK.

[6] Persson A. Stirna J. [Eds.], *The practice of enterprise modeling*. Springer, LNBIP 39, 2009.

[7] Sandkuhl K., *Improving engineering change management with information demand patterns*, in Proceedings of the International Conference on Product Lifecycle Management, Eindhoven, The netherlands, July 2011, Inderscience Enterprises.

[8] Freitag D., *Information extraction from HTML: application of a general machine learning approach*, in Proceedings of the 15th National Conference on AI, pp. 517–523, 1998.

[9] Jackson P., Al-Kofahi K., Tyrrell A., Vachher A., *Information extraction from case law and retrieval of prior cases*, in Artificial Intelligence, 150, 1–2, 239–290, 2003.

[10] Appelt D., Israel D., *Introduction to information extraction technology*, Tutorial for IJCAI-99, Stockholm, 1999.

[11] Degbelo A., Matongo T., Sandkuhl K., *Enterprise models as interface for information searching*, in P. Forbrig and H. Günther [Eds.], BIR 2010, LNBIP 64, Springer-Verlag Berlin Heidelberg, pp. 27–42, 2010.

[12] Smirnov A., Levashova T., Shilov N., Kashevnik A., *A hybrid technology for operational decision support in pervasive environments*, in L. Iliadis, I. Vlahavas, and M. Bramer [Eds.], Artificial Intelligence Applications and Innovations III, IFIP International Federation for Information Processing, Boston: Springer, 296, 3–12, 2009.

[13] *What is a Web service?*, in Web Services Architecture, W3C, 2004, http://www.w3.org/TR/ws-arch/.

[14] Feng Y., Liu X., Kerridge J., *An aspect-oriented component-based approach to seamless Web service composition*, in Intl. Transactions on System Science and Applications, Foreesight Academy of Technology, 5, 2., 116–125, 2009.

[15] Rao J., Su X., *A survey of automated Web service composition methods*, in Proc. of First Intl. Workshop on Semantic Web Services and Web Process Composition, SWSWPC 2004, San Diego, USA, pp. 43–54, 2004.

[16] Casati, F., Ilnicki S., Jin L., *Adaptive and dynamic service composition in EFlow*, in Proc. of 12th Intl. Conf. on Advanced Information Systems Engineering (CAiSE), Stockholm, Sweden, London: Springer Verlag, LNCS, vol. 1789, 2000, pp. 13–31.

[17] Schuster H., Georgakopoulos D., Cichocki A., Baker D., *Modeling and composing service-based and reference process-based multi-enterprise processes*, in Proc. of 12th Intl. Conf. on Advanced Information Systems Engineering (CAiSE), Stockholm, Sweden, London: Springer Verlag, LNCS, 1789, 247–263, 2000.

[18] Bubak M., Gubała T., Kapałka M., Malawski M., Rycerz K., *Workflow composer and service registry for grid applications*, in Future Generation Computer Systems, Elsevier, 21, 1, 79–86, 2005.

[19] Karakoc E., Kardas K., Senkul P., *A workflow-based Web service composition system*, in IEEE/WIC/ACM Intl. Confs. on Web Intelligence and Intelligent Agent Technology – Workshops, Hong Kong, China, pp. 113–116, 2006.

[20] Weise T., Bleul S., Comes D., Geis K., *Different approaches to semantic Web service composition*, in Proc. of Third Intl. Conf. on Internet and Web Applications and Services, ICIW 2008, Athens, Greece, pp. 90–96, 2008.

[21] Peer J., *Web service composition as AI planning – a survey*, Tech. Rep. Univ. of St. Gallen, Switzerland, http://elektra.mcm.unisg.ch/pbwsc/docs/pfwsc.pdf, 2005.

[22] Dorn J., Rainer A., Hrastnik P., *Toward semantic composition of web services with MOVE*, in Proc. of IEEE Joint Conf. on E-Commerce Technology (CEC'06) and Enterprise Computing, E-Commerce and E-Services (EEE'06), USA, San Francisco, IEEE CS Press, pp. 437–438, 2006.

[23] Oh S.-C., Lee D., Kumara S.R.T., *A comparative illustration of AI planning-based web services composition*, in ACM SIGecom Exchanges, New York: ACM, 5, 5, 1–10, 2006.

[24] Digiampietri L.A., Pérez-Alcázar J.J., Medeiros C.B., *AI Planning in Web services composition: a review of current approaches and a new solution*, in Proc. of XXVII Conf. Brazilian Comp. Sci. (CSBC2007), Rio de Janeiro, Brazil, pp. 983–992, 2007.

[25] Liu Z., Ranganathan A., Riabov, A., *A planning approach for message-oriented semantic web service composition*, in Proc. of 22nd National Conf. Artificial Intelligence, Vancouver, Canada, AAAI Press, 2, 1389–1394, 2007.

[26] Lecue F., Leger A., Delteil A., *DL reasoning and AI planning for Web service composition*, in Proc. of Intl. Conf. Web Intelligence and Intelligent Agent Technology – WI-IAT '08, Sydney, NSW, IEEE/WIC/ACM, 1, 445–453, 2008.

[27] Hu S., Muthusamy V., Li G., Jacobsen H.-A., *Distributed automatic service composition in large-scale systems*, in Proc. of Second Intl. Conf. on Distributed Event-Based Systems, Rome, Italy, 332, 233–244, 2008.

[28] Matskin M., Küngas P., Rao J., Sampson J., Petersen S.A., *Enabling Web services composition with software agents*, in Proc. of Ninth IASTED Intl. Conf. on Internet and Multimedia Systems, and Applications (IMSA 2005), Honolulu, Hawaii, USA, ACTA Press, pp. 93–98, 2005.

[29] Buhler P.A., Greenwood D., Weichhart G., *A multiagent Web service composition engine, revisited*, in Proc. of IEEE Joint Conf. (CEC/EEE 2007) on E-Commerce Technology (9th CEC'07) and Enterprise Computing, E-Commerce and E-Services (4th EEE'07), Tokyo, Japan, IEEE Computer Society, pp. 529–532, 2007.

[30] Falou M.E., Bouzid M., Mouaddib A.-I., Vidal T., *Automated Web service composition: a decentralised multi-agent approach*, in Proc. of IEEE/WIC/ACM Intl. Conf. on Web Intelligence and Intelligent Agent Technology, IEEE Computer Society, 1, 387–394, 2009.

[31] Lind M., Seigerroth U., *A Multi-Layered Approach to Business and IT Alignment*, Hawaii International Conference on System Sciences (HICSS43), Hawaii, January 5–8, 2010.

[32] Pepiot G. et al., *UECML: Unified Enterprise Competence Modelling Language.* Computers in Industry, 58, 130–142, 2007.

[33] Kopecky J., Vitvar T., Bournez C., Farrell J., Sawsdl: Semantic annotations for wsdl and xml schema. Internet Computing, Vol. 11, Issue: 6, Nov.–Dec. 2007, ISSN 1089-7801, IEEE Computer Society, 2007.

[34] Vitvar T., Kopecky J., Viskova J., Fensel D., *WSMO-lite annotations for web services*, In: The 5th Annual European Semantic Web Conference (ESWC 2008), 1–5 Jun 2008, Tenerife, Spain, 2008.

[35] Rajasekaran P., Miller J., Verma K., Sheth A., *Enhancing Web Services Description and Discovery to Facilitate Composition*, Proceedings SWSWPC at IEEE International Conference on Web Services (ICWS 2004), San Diego, California, USA, July 2004.

[36] Alexander Smirnov, Alexey Kashevnik, Nikolay Teslya, Nikolay Shilov, Andreas Oroszi, Mario Sinko, Michael Humpf, Jens Arneving, *Knowledge Management for Complex Product Development: Framework and Implementation*, Proc. of the 10th International Product Lifecycle Management Conference – PLM13, July 06–10, Nantes, France, 2013.

[37] Oroszi A., Jung T., Smirnov A., Shilov N., Kashevnik A., *Ontology-Driven Codification for Discrete and Modular Products*, In: International Journal of Product Development, Inderscience, 8, 2, 162–177, 2009.