Context-aware Knowledge Management for Socio-Cyber-Physical Systems: New Trends towards Human-machine Collective Intelligence

Alexander Smirnov[a], Nikolay Shilov[b] and Andrew Ponomarev[c]
SPIIRAS, 14th Line, 39, St.Petersburg, Russian Federation

Keywords: Socio-Cyber-Physical Systems, Collective Intelligence, Hybrid Systems, Context-aware Knowledge Management, Ontology-based Systems, Multi-aspect Ontology, Role-based Organization, Dynamic Motivation.

Abstract: The competitiveness of companies and organizations heavily depends on how they maintain and access highly decentralized up-to-date information & knowledge coming from various resources located in their Socio-Cyber-Physical Systems. Such systems tightly integrate heterogeneous resources of the physical world and IT (cyber) world together with social networking concepts. Context-Aware Knowledge Management is becoming de facto one of the required business strategies in these systems. Its goal is to facilitate knowledge transfer and sharing in the context of business structures and activities bound together with the cultural norms. This keynote presents new trends (including role-based organization, dynamic motivation mechanisms and multi-aspect ontology) in knowledge management for socio-cyber-physical systems. Such trends can facilitate creation of innovative IT & HR environments based on human-machine collective intelligence, where information & knowledge are shared between participants and across collectives of participants, who can be both people (collective intelligence as the methods used by humans to act collectively for problem solving) and software services (based on artificial intelligence models). The keynote considers examples of trends and their implementation experience in a global production company.

1 INTRODUCTION

The concept of Socio-Cyber-Physical System (SCPS) integrating in real-time physical systems (e.g., physical production equipment, vehicles, devices), IT components (e.g., enterprise resource planning, manufacturing execution systems or other information systems), and human actors (organizational roles and stakeholders) at individual and social network level is becoming more and more important in understanding modern IT landscape.

Currently, more and more systems (including "system of systems") in many areas are recognized to be socio-cyber-physical, and this spurs on the research in the area of SCPSs, aimed at creating coherent tools and methodologies for the SCPSs development and evolution. Quite a few good SCPSs examples can be found in modern production environments, especially those adopting the Industry 4.0 concept.

Advances in the mobility, cloud computing, crowdsourcing, and big data analytics increase the number and kinds of networked connections in business environments, as well as the opportunities for people and machines to derive unpredictable value from these connections (Pew Research Center, 2014).

Knowledge management (KM) allowing to locate knowledge/skill for a task at hand is a crucial for successful collaboration, in particularly in the systems with heterogeneous entities (as in SCPSs). Distributed work in product design, manufacturing, and supply management projects requires decision support for the involved parties tailored to the actual context of these parties (depending on their nature).

[a] https://orcid.org/0000-0001-8364-073X
[b] https://orcid.org/0000-0002-9264-9127
[c] https://orcid.org/0000-0002-9380-5064

Smirnov, A., Shilov, N. and Ponomarev, A.
Context-aware Knowledge Management for Socio-Cyber-Physical Systems: New Trends towards Human-machine Collective Intelligence.
DOI: 10.5220/0010171800050017
In *Proceedings of the 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2020) - Volume 3: KMIS*, pages 5-17
ISBN: 978-989-758-474-9

Network-wise modern SCPSs are based on integration of a number of networks supported by the following information technologies (A. Smirnov & Sandkuhl, 2015):

- Social networks: who knows whom => Virtual Communities;
- Knowledge networks: who knows what => Human & Knowledge Management;
- Information networks: who informs what => Internet/Intranet/Extranet/Cloud;
- Work networks: who works where => Decision Support based on Crowdsourcing and Recommendation Systems;
- Competency networks: what is where => Knowledge Map;
- Inter-organizational network: organizational linkages => Semantic-Driven Interoperability.

In general, SCPSs are reconfigurable dynamic systems; their elements may have variety of possible states and arrange in dynamically arrange in problem-centric compositions. This provides an additional requirement for successful KM in SCPSs. Namely context-awareness. The context is usually defined as any information that can be used to characterize the situation of an entity, where an entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves (Dey, Abowd, & Salber, 2001).

This paper describes some trends in implementing context-aware KM in SCPSs.

The rest of the paper is organized as follows. Section 2 describes some important trends in KM in SCPSs. Section 3 discusses practical application of these trends in solving KM problems in a production company. Finally, section 4 presents a design of a human-machine collective intelligent environment, which follows these trends and can be used in a variety of problem domains to effectively solve KM problems at decision support by human-machine collectives.

2 MODERN TRENDS IN CAKM FOR SCPS

This section describes some modern trends in the context-aware knowledge management (CAKM) for socio-cyber-physical systems and shows how the respective emerging technologies can facilitate the creation of innovative IT&HR environments.

Ontology-based knowledge representation is in the core of these trends; it is their enabler. The purpose of ontologies is to represent knowledge about a certain domain in a machine-readable way. Ontologies allow to describe, share and process knowledge considering its syntax along with its semantics. They are formal conceptualizations of certain domain of interest that are shared between different applications (Gruber, 1993; Staab & Studer, 2009). The ontology describes concepts, their relationships and axioms thought to exist in the given domain. They are considered an efficient mean to solve the interoperability problem. In particular, ontologies turn out to be effective in encoding context.

Context model serves to represent the knowledge about a current situation (the environment properties, the current problem, as well as states of the stakeholders).

These models, for instance, are used to reveal user preferences based on the analysis of the context representations in conjunction with the implemented decisions.

2.1 Role-based Organization

Personalized support is important for modern business applications. As a rule, it is based on application of the profiling technology. Each user (a human or an information system) works on a particular problem or scenario represented via a context that may be characterized by a particular customer order, its time, requirements, etc.

Research efforts in the area of information logistics show information and knowledge needs of a particular employee depend on his/her tasks and responsibilities (Lundqvist, 2007). Therefore, in business applications the idea of personalization (identification of implicit context of the request) can be extended with the knowledge of the user's role. Besides, it is also the case that representatives of adjacent (in terms of business process) roles can have slightly different goals and use different terminology (even referring to the same concepts).

The idea of the role-based approach is to consider the workflows and information models from perspectives of different roles that deal with them.

Role-based organization for ontology-based KM assumes the following steps:

1. Structural information about workflows and the problem domain is collected and described in the common ontology.
2. User roles are identified and their relevant parts of the common ontology are defined.

3. Tasks assigned to the identified roles are defined.
4. Knowledge required for performing identified tasks is defined.
5. Based on the identified roles, tasks and knowledge new knowledge-based workflows are defined.
6. Corresponding role-based knowledge support of the workflows is provided based on the usage of the common ontology and knowledge / information storages.

This process repeats for each particular role, with some knowledge being reused by several roles.

The implementation of the approach is described in Section 3.1.

2.2 Dynamic Motivation Mechanisms

Despite the prevalence of the KM systems aimed at improving knowledge sharing within the organization sometimes these KM mechanisms add responsibilities and activities that have to be done by employees and that are not seen as important as the primary (productive) activities. Therefore, employees may evade using the organized KM solutions or even feel threatened by organizing their knowledge in an accessible manner as it might make them 'replaceable'. An important task of management is to establish open and fair corporate culture that values KM. One of the most important aspects that have to be considered is aligning organization and employees goals via employee motivation (Friedrich, Becker, Kramer, Wirth, & Schneider, 2020). Especially, dynamic motivation (the type of motivation that changes within a short period of time). A good example of dynamic motivation used by the retailer companies are: the best sellers boards, scores in the corporate systems and etc.

The empirical study has proven that dynamic motivation seems to yield high levels of engagement, learning, and of performance and effectiveness in organizational implementation processes. In addition, dynamic motivation also seems to positively contribute to collaborative work and team performance (Ferreira, Araújo, Fernandes, & Miguel, 2017).

The use of dynamic motivation in some SCPS relies on answering two questions:

1) How should the participants be motivated (what rewards are effective)?
2) What software solutions can be used to define dynamic motivation mechanisms?

The first question is extensively studied in human resources management area.

The second one is more relevant to IT. There are two classes of solutions: specialized solutions (tailored for the particular problem) and generic solutions.

An example of using specialized solution for implementing the dynamic motivation approach to increase the project management efficiency based on the competency management system is described in (Smirnov, Kashevnik, et al., 2019). The solution includes reference and mathematical models of language expert network, which are used for the automated assignment of organization's personnel to projects. They allow formalizing not only the individual employees' skills, but also their achievements and strengths.

A prominent example of generic solution is PRINGL language (Scekic, Truong, & Dustdar, 2015) allowing to define motivation policies in an application independent way and connect to some information system via an application programming interface.

2.3 Multi-aspect Ontology

The purpose of ontologies is to represent knowledge about a certain domain in a machine-readable way. However, in some complex domains, like Product Lifecycle Management (PLM), the application of ontologies is complicated since it has to deal with interdisciplinary information and knowledge related to different phases (Shilov, Smirnov, & Ansari, 2020). The terminology and notations used in various processes are different since they are aimed at solving tasks of different nature that require different techniques (Asmae, Souhail, Moukhtar, & Hussein, 2017; Palmer, Urwin, Young, & Marilungo, 2017). To a certain extent, this problem is similar to that of role-based information representation, where the information and knowledge have to be presented to different roles in different views and terminologies.

Some research efforts were aimed for enriching ontologies with additional information that could represent additional facts originally described in a different notation (e.g., semantic annotations (Liao, Lezoche, Panetto, & Boudjlida, 2016), DAML+OIL extensions for configuration problem descriptions (Felfernig, Friedrich, Jannach, Stumptner, & Zanker, 2003), and others). However, this still cannot be an efficient solution for problems of integrating information and knowledge from multiple different notations and terminologies, which is the case for PLM.

One of the common solutions for multi-domain systems is having a common ontology at the top and its extension for specific sub-domains (e.g., configuration problem solving). However, it is not efficient for dynamic domains with large number of sub-domains, since this would require a continuous ontology matching and modifications of the common ontology.

Ontology matching can also be used separately for establishing links between multiple domain-specific ontologies. However, manual ontology matching would require too much time and efforts in dynamically changing domains and automatic ontology matching is still not a reliable instrument since the existing methods deliver high level of precision only in narrow domains.

The authors of (Lafleur et al., 2016) propose a model-driven interoperability framework aimed at supporting relationships between products and manufacturing equipment. They form a "connection framework" describing relationships between different product ontologies maintained in the PLM system and different ontologies of manufacturing capabilities managed in the Manufacturing Process Management system. However, having multiple ontologies for different tasks is not an efficient solution for the problem identified either. Since translating information from one specific ontology to another assumes a translation between the source ontology and the common ontology and then between the target ontology, what eventually will cause information losses.

Another approach is to preserve the original domain ontologies and build an additional layer at the top of them. The authors of (Hagedorn, Smith, Krishnamurty, & Grosse, 2019) propose to use a Basic Formal Ontology (BFO) as a top-level ontology for describing various engineering domains, and to re-engineer the existing ontologies so that they would be compliant to it.

Viewing a problem domain from a number of viewpoints has resulted in appearance of Multi-Viewpoints Ontology (MVpOnt). In MVpOnt each viewpoint corresponds to the knowledge representation model useful for a particular task, process, or a group of people co-existing in a common information environment and sharing some information and knowledge. These viewpoints are described in a specialized language for the multi-viewpoint ontologies called MVP-OWL) (Hemam & Boufaïda, 2011). In 2018 MVP-OWL was extended with probabilistic reasoning support (Hemam, 2018).

MVP-OWL extends OWL-DL (the complete description is presented in (Hemam & Boufaïda, 2011)). Firstly, it supports viewpoints that describe information and knowledge related to a certain task or process. Secondly, classes and properties are divided into two groups: local – observed only from one viewpoint, and global – observed from two or more viewpoints. The instances can only be local, however since MVP-OWL supports multi-instantiation, the instances can exist in several viewpoints at the same time. Thirdly, the authors introduce "bridge rules" of four types, which enable relating concepts from different viewpoints.

This approach is the most suitable for the problem set since it supports resolving terminological issues, and also makes it possible to preserve original formalisms used in existing ontologies.

3 CAKM IMPLEMENTATION

This section describes several particular organizational KM problems and how they are successfully approached by modern solutions described in Section 2.

The problem at hand is product and knowledge management in a large automation manufacturing company. This section integrates results of several projects carried out by the research team, the paper authors belong to, together with representatives of the company.

3.1 Role-based Organization

This approach was implemented in the frame of the project reported in (Smirnov, Levashova, & Shilov, 2015).

The first step of the approach implementation was the ontology creation. The resulting ontology consists of over 1000 classes organized into a four level taxonomy based on the VDMA (Verband Deutscher Maschinen – und Anlagenbau, German Engineering Federation) classification (*VDMA. German Engineering Federation*, 2018). Later it was extended with descriptions of complex products, their components and compatibility rules.

At the second step, the major roles, whose workflows were addressed by KM implementation, have been identified. They included product manager, product engineer, production manager, and production engineer.

Then, at steps 3 and 4, their tasks and knowledge/information needs were analysed. For example, the product manager works with customers and their needs. Since the terminology used by customers differs from that used by product engineers, a mapping between the customer needs and internal product requirements had to be established.

At steps 5 and 6 the knowledge-based workflows were defined, and corresponding supporting tools were built.

The project showed that such approach enabled implementing KM incrementally, with initiative coming from employees. E.g., an experimental knowledge-based support of one workflow could be implemented for one user role letting the users estimate its efficiency and convenience. Then, workflows reusing some knowledge of the experimental workflow can be added, etc. Representatives of other roles seeing the improvements of the implemented knowledge-based workflows also wish to join and actively participate in the identification of the knowledge needed for their workflows and further turning their workflows into the knowledge-based ones.

3.2 Dynamic Motivation

An example of successfully leveraging dynamic motivation is automating and facilitating a translation process involving company employees from different countries (A. Smirnov, Kashevnik, et al., 2019). The translation process was implemented as a distributed network of language experts, and dynamic motivation was leveraged to incentivize experts (found by maximization of the global fitness function) to take part in the translation.

An example of the generated skill tree that is used to describe expert's competence profile as well as task requirements is presented in Fig. 1. The skill tree for the developed language experts network consists of three main parts: dictionary, industry segment, and technical area that describes the mentioned problem domains.

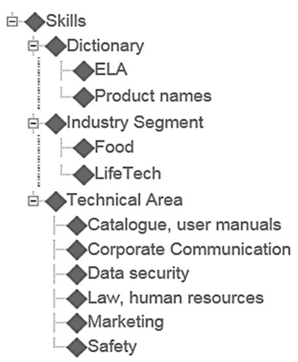

Figure 1: Example of skill tree.

Every expert is described by a competence profile. The expert profile contains: information about the expert, list of competencies, and professional assessment (global skill level, GSL). Global skill level is calculated based on a number of successfully completed tasks this expert performed, his/her availability estimation for task performing, estimation of how long the expert works in the company, qualification of the expert, and rewards the expert received from the manager.

For the definition of the proofreading task it is proposed to use the following structure (see Table I). The task form accessible to the expert includes the task structure presented in the table as well as the task discussion interface that allows proofreaders to exchange their knowledge about it.

Table 1: Proofreading task description.

Name	Description
Due Date	Date when the task should be performed
Source Language	Source language of the term
Target Language	Target language of the term
Term	Term to be translated
Translation	Translation made by translation agency
Task Context	Context that helps an expert to perform the translation. It includes the project where the translation will be used, "in sentence context", technical area, industry segment, and etc.

The list of possible motivations used in the system includes two main groups of motivations: material and non-material. Every motivation is specified by budget, value, and monetary benefit as well as it can be supported globally or only by one or several local companies.

For example, an expert can be motivated by a shopping voucher (20 EUR). In this case spent budget will be 20 EUR as well as monetary benefit that determines the value of this present for the expert. At the same time the value of a positive recommendation to the expert's boss could be evaluated as 10 (maximum value) but budget and monetary benefit is 0, since the company does not spend money on it.

The system also provides a number of forms for managing rewards. First, a form to display the list of rewards assigned to each expert (including date/time of the assignment) and define new rewards (Fig. 2). Using this form, a manager can select an expert(s) and reward. The system shows the left monetary benefit for each expert in the current year.

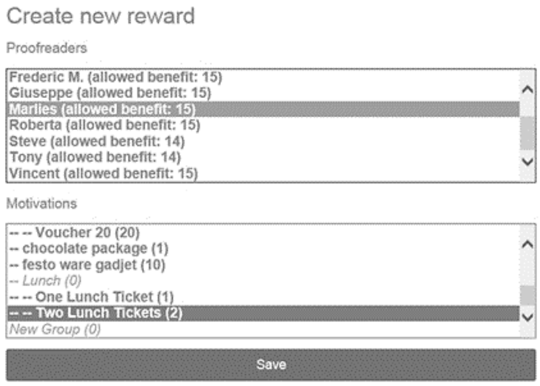

Figure 2: Example of the new reward definition.

3.3 Multi-aspect Ontology

Several projects carried out for the same production company have led to a necessity to share information and knowledge between several workflows and departments that do not share the same terminology. Besides, different tasks of different workflows required application of different formalisms what resulted in a necessity of developing a multi-aspect ontology (A. Smirnov, Shilov, & Parfenov, 2019). These different views can be successfully synchronized and matched with a help of multi-aspect ontology, being a formalized instrument supporting various processes of the considered company. For this reason, the ontology had to cover processes, that were addressed during development of the information and knowledge management systems. As an illustrative example for this paper the aspects of "*Product Engineering*", "*Sales*", and "*Strategic Planning and Production*" that correspond to different PLM phases have been selected.

Development of the aspect ontologies can be done on the basis of any existing methodology of ontology development, e.g., METHONTOLOGY (Fernández-López & Gómez-Pérez, 2002). Aspect ontology can be also built using a different methodology since the aspects are independent. When developing an aspect ontology, a reuse of existing ontologies is beneficial (e.g., typical subproblems usually already have established ontologies) though not obligatory.

The illustration of the developed multi-aspect ontology is given in Fig. 3. The ontology was built based on the top-level ontology presented in (Borsato, Estorilio, Cziulik, Ugaya, & Rozenfeld, 2010). Earlier developed ontologies for different tasks have been used as the aspects (one task corresponds to one aspect). For the illustration purposes the aspects with different formalisms have been selected. Below, each of them is described with corresponding references.

The first considered aspect *Product Engineering* describes the task of definition of a new product and its features (Oroszi, Jung, Smirnov, Shilov, & Kashevnik, 2009), which is currently done in the NOC tool. This aspect is defined in OWL. The goal of this task is definition of new products and product families with their possible characteristics by a product engineer. During this process, the product engineer has to make sure that the defined products and characteristics are consistent (the Pellet reasoner is used for this purpose). The sample classes presented in the figure include "*Product Family*" (high level generalization of products), "*Product Group*" (lower level generalization of products, a subclass of *Product Family*), "*Product*" (simple or modular product, a subclass of *Product Group*), and "*Feature*" (product characteristics, associated with the class *Product*).

The second considered aspect is *Sales*. It describes the task of defining and using constraints between product characteristics and product combinations in an assembly. Definition of the constraints is done via the CONSys tool by product manager, and their usage is done in the CONFig tool by a customer or product/solution managers (A. V. Smirnov, Shilov, Oroszi, Sinko, & Krebs, 2018). For the purpose of constraint satisfaction technology support, the formalism of object-oriented constraint networks was used. The example classes from this aspect are "*Product*" (can be a product or a product combination), "*Parameter*" (parameter of a product, e.g., "*mass*", "*power*", that can match product characteristic but it is not always the case), and "*Constraint*" (mathematical constraints limiting or calculating values of product characteristics depending on other characteristics).

The third presented aspect is *Strategic Planning and Production*. The task solved in this aspect is definition of strategy regarding production classes. Three classes are considered: "*ETO*" (engineered to order, longest lead time), "*ATO*" (assemble to order, medium lead time), and "*PTO*" (pick to order, shortest lead time) (A. V. Smirnov et al., 2018). Solving this task is based on pre-defined rules, and, hence it is defined as a set of classes and production rules ("if … then …"). Based on these rules the lead times and production plants for the products are defined. Example classes of this aspect are "*Production Class*" (the superclass for the above mentioned "*ETO*", "*ATO*", and "*PTO*" classes), "*Product*", and "*Plant*".

To sum up the following elements of the multi-aspect PLM ontology have been defined:

Aspects: Product Engineering, Sales, Strategic Planning and Production.

Local Classes (by aspect):

Product Engineering aspect: Product Family, Product Group, Product, Feature.

Sales aspect: Product, Parameter, Constraint.

Strategic Planning and Production aspect: Product, Production Class, Plant, Rule.

Global level has the following classes: Thing, Attribute, Product, Dependency, Group, Resource.

To establish connections between aspects and the global level, **bridge rules** have been defined. As an example, the bridge rules of bidirectional inclusion (symbol $\overset{\equiv}{\leftrightarrow}$), meaning that two concepts from different aspects are equal, for the class Product are presented:

Product $\overset{\equiv}{\leftrightarrow}$ $\text{Product}_{\text{ProductEngineering}}$;

Product $\overset{\equiv}{\leftrightarrow}$ $\text{Product}_{\text{Sales}}$;

Product $\overset{\equiv}{\leftrightarrow}$ $\text{Product}_{\text{StrategicPlanningAndProduction}}$

i.e., the concept product in different aspects has the same meaning.

The resulting ontology made it possible to establish links between heterogeneous information models, and, for example, changes made in the *Product Engineering* task can be easily reflected in the *Sales*.

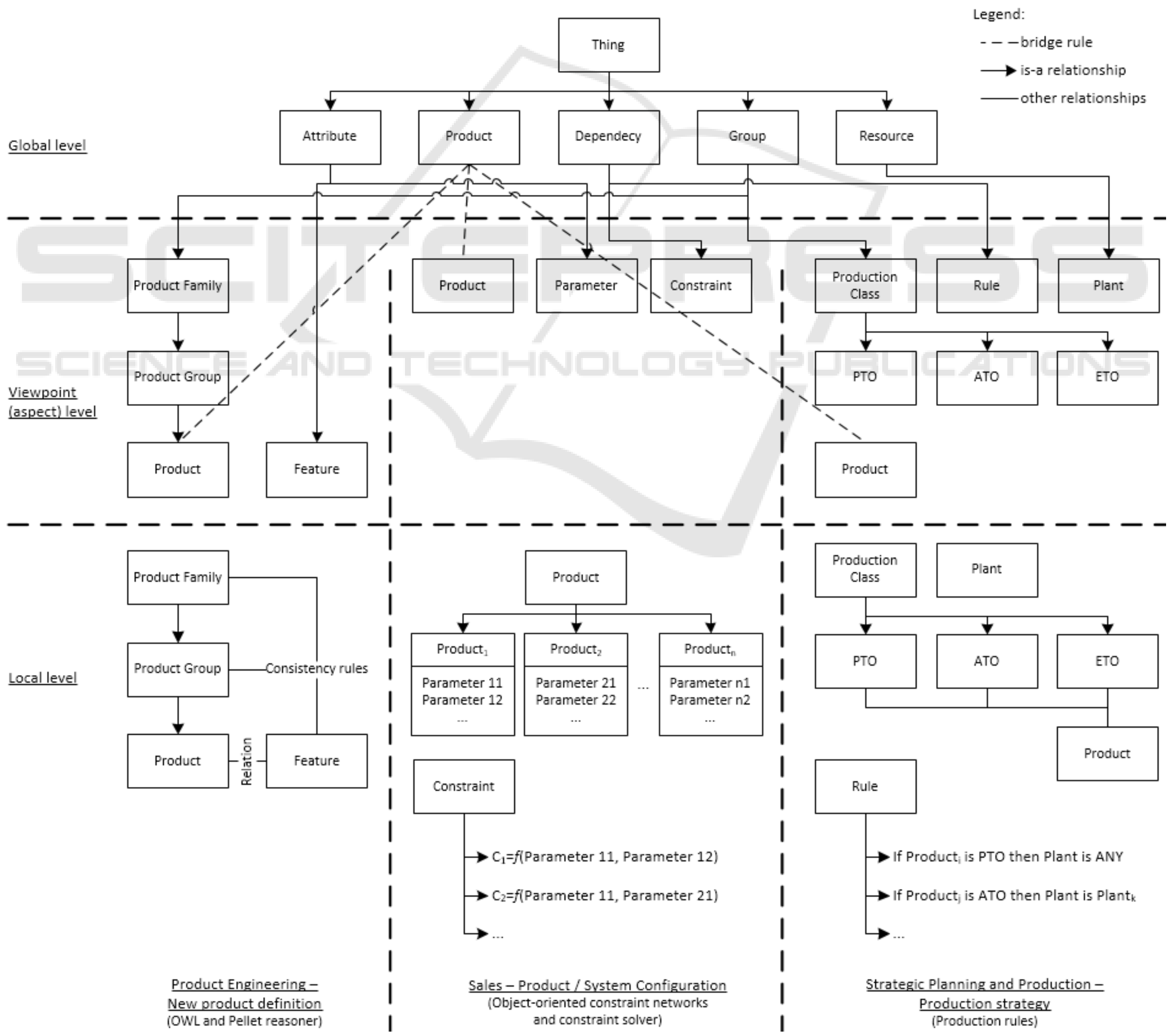

Figure 3: Multi-aspect ontology for three aspects.

4 CONCEPT OF HUMAN-MACHINE COLLECTIVE INTELLIGENCE ENVIRONMENT

The experience of implementing the above novel techniques for CAKM in a production environment and benefits they brought have led to an idea that they could be applied in a more general way to create an environment supporting human-machine collective intelligence.

The problem of human-machine collaboration and collective intelligence in particular have attracted attention of researchers in several perspectives and have posed a number of important questions (Jennings et al., 2014).

The proposed human-machine collective intelligence environment is based on the following foundations:

- One of the established facts about collaborative work on complex problems is that it requires certain agent autonomy and self-organization (Retelny, Bernstein, & Valentine, 2017).
- To achieve interoperability between human and software participants, the environment should support some structured representation of the discourse contents and/or task distribution. A good example is the Dicode project implemented within the framework of the European FP7-ICT program (Karacapilidis & Tampakas, 2019), proposing an ontological presentation of the argumentation process and a number of visual tools for working with a formalized set of interrelated arguments.

In this research, however, an environment is built where heterogeneous agents (human and software) would be able to collectively decide on the details of the workflow. Its distinguishing features are:

- The support for self-organization (in contrast to pre-defined workflows);
- Flexible role-based distribution of responsibilities;
- The use of ontologies (and, in particular, multi-aspect ontologies) to support human-machine interoperability and knowledge management.

The purpose of this environment is to implement basic discovery, information exchange and organization routines to allow agents of different nature (human and software) to collectively tackle organizational decision-making problems.

The primary goal is to support cooperation of relatively short-lived (hours to several days) *ad hoc* teams. Another limitation is that the environment is inherently dedicated to decision support problems. Therefore, the design is influenced by decision-making methodologies and the workflow implemented by a team mostly corresponds to a typical decision-making process.

There are following principal actors differentiated by the environment design: end-user (decision-maker), participant, and service provider (Fig. 4). End-user (decision-maker) uses the environment to get help in making a decision. He/she describes the problem and posts it so that the problem description is visible to a specified community. Participant is an active entity (human or a software service) working on a problem given by the end-user. Finally, a service provider develops, integrates to the environment, and supports software services that can act as participants working on some problem given by the end user. Service provider is also responsible for the deployed services, assuaging the problem of service accountability.

Core entities involved in most of the environment processes are the *problem* and the *team*. Problem is introduced by an end-user and then is addressed by the participants' team. The problem description has a complex structure and representation. First of all, it contains information, specified by the end-user (initial statement), and also includes all information produced by the team. So, during team's activity the problem becomes more and more detailed. Second, to enable (at least, partially) an effective interpretation by software agents the problem description is represented in a semi-structured way. In particular, machine readability is achieved via using ontologies. To facilitate the use of ontologies for people the environment makes it as implicit as possible by relying on three techniques:

- Implicit ontological representation of the structure of problem information.
- Natural language processing. Using advances in this area it is possible to infer the role of some information pieces, its relationship with the goal and/or some line of argumentation and so on.
- GUI-based nudging participants to encode problem structure in an ontology-compatible way.

The environment defines two basic ontologies, representing different aspects of the collaborative decision support (Figure 5):

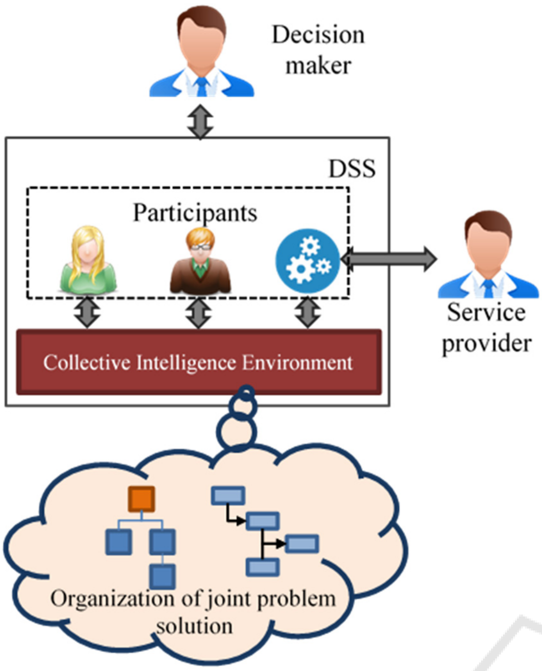

Figure 4: Principal actors.

- Decision-making ontology. This ontology defines main concepts that are used during decision-making (criterion, alternative, evaluation etc.) and interaction between them. The ontology is based on the analysis of existing decision-making methodologies and has been built to support majority of them.
- Collaboration and coordination ontology. It defines the concepts used in distributing work among team members (role, responsibility, dependency etc.).

The use of above ontologies allows artificial agents to 'understand' the processes taking place in the team and contribute to them. However, for the ontology-based decision-support agents, there also exists a possibility to define an application ontology and to map it to the decision-making ontology. By this process some parts of the problem situation become connected to the general decision-making terminology.

The way problem information becomes richer and grows via interaction of agents, to some extent resemblances to stigmergy (Heylighen, 2016) and intelligent systems based on the blackboard interaction principle.

Another already mentioned core entity is *the team*. The team in the context of the environment is defined as a heterogeneous group (consisting of human participants and software services) working towards solution of a particular *problem*. Each problem has a team dedicated to it. Obviously, a participant may be a member of several teams, or not be a member of any team.

Initial team formation is based on the same principles used in most of the crowdsourcing platforms and knowledge networks (Ahmad, Battle, Malkani, & Kamvar, 2011): each participant has a profile describing key specializations, problem-solving history, as well as the history of previous collaborations (with mutual evaluations). There is a massive list of publications why each of this components of the profile is necessary and how it affects the efficiency of teaming. The initiative in this process is mixed in the sense that a contributor should send a proposal to the end-user, consisting of one or more team members (proposal may include several participants that already have some positive experience of working together), and end-user has to collect the initial team. However, decisions of the both parties – participants and end-users – are assisted by environment. The participants may choose to receive recommendations if some problem touching his/her area of competences is posted. On the other hand end-users may explore the description and history of all the participants mentioned in the proposals.

Due to much uncertainty typically associated with decision-making, it is often the case that during the work on the problem, the team understands that it lacks some competencies or resources. Therefore, the team may create a new resource requirements, that are registered in the environment and resolved in a manner, similar to the initial team formation process (participants have to actively apply for the positions in the team, however, both sides are assisted by the environment mechanisms).

It should be noted, that it does not fully apply to the software participants (services). As the throughput of software services is not as limited as the throughput of humans, and the execution is relatively cheap, software services are passively connected to any team and by the mechanisms of the environment (ontology-based publish-subscribe) are watching the processes taking place with the problem. There are two states a software service can be in w.r.t. the team: dormant and active. Initially, all services are in the dormant state and are waiting for specific conditions during the problem-solving. If these conditions defined by a particular service are met, the service tries to activate, describing its purpose and terms of use. If the team agrees that the service is useful for the problem, the service is allowed to activate (change state to active) and become a member of the team. Otherwise, the

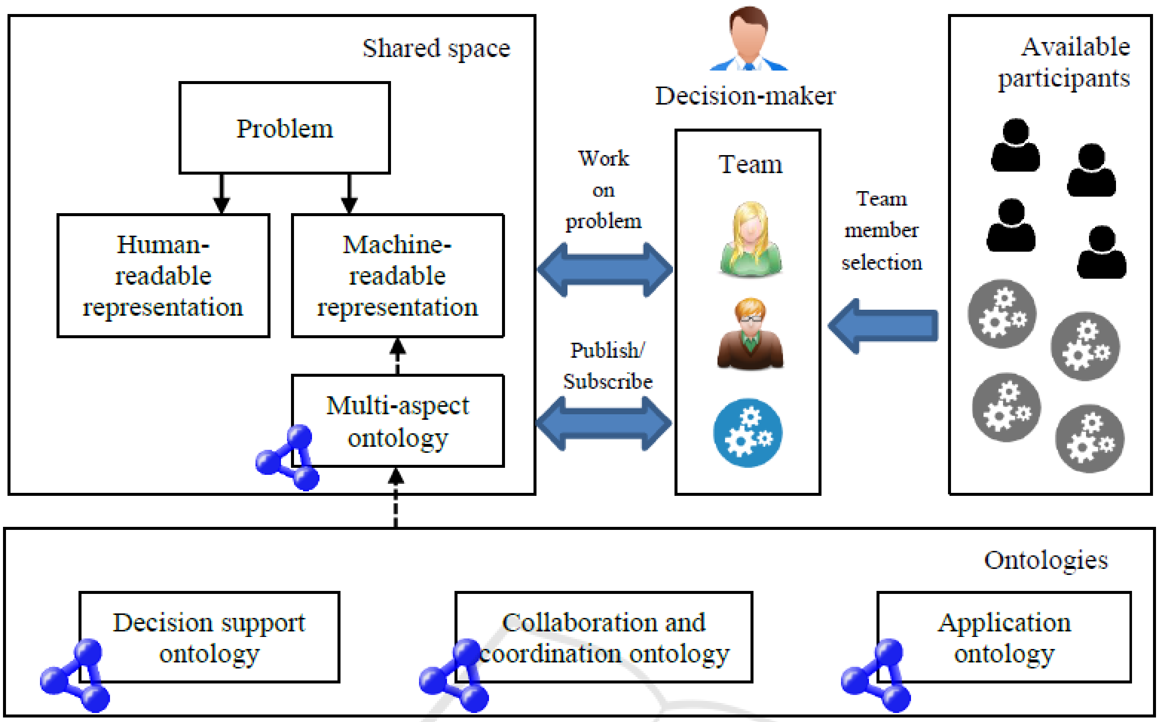

Figure 5: Conceptual model of the environment.

service remains dormant. Active service may also be transferred to the dormant state by a decision of the team. Besides, the services can be accessed via a service catalogue and activated manually by team members.

Active services can be used by the team members. The mechanics of their usage depends on the service's kind. There are two main types of services:

- Problem-solving service;
- External tool and database access service.

Problem-solving service accesses the problem information described in the form of ontology and natural text, and can actively add information pieces to it. An example of such service is a statistics-based question answering service – if it detects a question about some facts (e.g., "How many people die from tuberculosis in the World in one year?") and can answer it in some form, it adds an answer to the question. Another example is a service that derives from the problem information a current set of alternatives and their evaluations, builds a Pareto optimal set and adds it to the problem information.

External tool and database access services in their activated form only provide an access to a specified resource. For example, if the team needs an epidemic database, it can activate the service that grants access to this database and use it for queries.

Simultaneously two processes take place when team works on a problem: *solution preparation* and *decision support (re)organization*. Both of these processes are supported by mechanisms provided by the environment. Solution preparation is main productive process, during which problem is enriched with new information and artefacts created by team members. The result of this process is fully detailed description of a problem situation, weighted alternatives and their estimated consequences – accepted by the end-user. Decision support (re)organization process represents all the activities aimed at planning and organization of team work (e.g., deciding whether additional resources are required, assigning team member responsibilities, setting task deadlines and identifying new tasks to be solved in order to reach the goals of the whole process).

Several ontologies used to describe the current problem state are connected by the multi-aspect ontology approach. Two main aspects used in describing the problem are decision support aspect and domain aspect. For example, if the environment is used in a smart tourism scenario to select a tourist itinerary, then possible alternatives to consider and evaluate are tourist itineraries. Therefore, class *Alternative* of the decision-making aspect in this problem setting is connected with the *Itinerary* concept of the domain aspect. Further, evaluation of the itineraries done by the team can be interpreted as evaluations of the alternatives which is an essential step in making a decision. It should be noted, that mutual mapping of the aspects is realized in the context of the problem (in location problems Alternative is mapped to geographic point, etc.).

By providing a set of mechanisms (maintaining the problem state, discourse logics, team formation processes etc.) the environment supports the activities of the human-machine team. One important specific mechanism provided by the environment is soft guidance by offering situation-specific cooperation patterns to the team. In this sense, the environment plays the role similar to the facilitator in group decision support systems. These recommendations are based on the number of identified collaboration patterns: generate, reduce, clarify, organize, evaluate, build consensus. These patterns form basic activities performed by members of the team. Sometimes, current activity is defined explicitly during decision support (re)organization process (e.g., there might be an alternative generation activity). In other cases, pattern can be recognized by certain structure in the ontological description of the problem state (e.g., if two or more participants offer different estimations for the same alternative, then these estimations should be reconciled during consensus building). An important role of these patterns is that they allow to structure the team activities (w.r.t. the goal) and tie existing methods to these activities. For example, if it has been recognized, that the team needs to build consensus on the set of the alternatives estimation, a number of methods for building consensus could be recommended.

It can be seen, that the design of the environment is heavily affected by the discussed trends. First of all, ontology-based context modeling and specialization are at the core of the problem representation. The problem situation and all the artefacts are represented in the form of ontology, which allows to achieve interoperability between human and software participants. Role-based organization and multi-aspect ontologies are used to help to reconcile different aspects of the decision support (e.g., domain structure vs. process structure), besides, role-based organization is used also in the foundational layer, because every decision-making process in a coarse decomposition can be viewed as an interaction of different roles (project leader, data analyst, domain expert, etc.), and the environment supports the definition of the team via set of roles. Finally, dynamic motivation mechanisms play a role in process planning and team recruiting, because reward sharing is an important aspect of process definition.

5 CONCLUSIONS

The paper discusses three modern trends in context-aware knowledge management for socio-cyber-physical systems. In particular:

- role-based organization,
- dynamic motivation mechanisms, and
- multi-aspect ontology.

Each of these trends (or their combination) can be implemented in a variety of systems, improving the effectiveness of knowledge eliciting, storing, and utilization, which can have a major positive impact on the effectiveness of the whole system (organization).

The paper also introduces a concept of human-machine collective intelligence environment, making use of all these trends.

ACKNOWLEDGEMENTS

The research is partially funded by the Russian State Research, project 0073-2019-0005. The research on the human-machine collective intelligence for decision support is funded by the Russian Science Foundation, project 19-11-00126.

REFERENCES

Ahmad, S., Battle, A., Malkani, Z., & Kamvar, S. (2011). The jabberwocky programming environment for structured social computing. *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology - UIST '11*, 53–64. https://doi.org/10.1145/2047196.2047203

Asmae, A., Souhail, S., Moukhtar, Z. El, & Hussein, B. (2017). Using ontologies for the integration of information systems dedicated to product (CFAO, PLM…) and those of systems monitoring (ERP, MES□). *2017 International Colloquium on Logistics and Supply Chain Management (LOGISTIQUA)*, 59–64. https://doi.org/10.1109/LOGISTIQUA.2017.7962874

Borsato, M., Estorilio, C. C. A., Cziulik, C., Ugaya, C. M. L., & Rozenfeld, H. (2010). An ontology building approach for knowledge sharing in product lifecycle management. *International Journal of Business and Systems Research*, *4*(3), 278. https://doi.org/10.1504/IJBSR.2010.032951

Dey, A. K., Abowd, G. D., & Salber, D. (2001). A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human–Computer Interaction*, *16*(2–4), 97–166. https://doi.org/10.1207/S15327051HCI16234_02

Felfernig, A., Friedrich, G., Jannach, D., Stumptner, M., & Zanker, M. (2003). Configuration knowledge representations for Semantic Web applications. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *17*(01), 31–50. https://doi.org/10.1017/S0890060403171041

Fernández-López, M., & Gómez-Pérez, A. (2002). Overview and analysis of methodologies for building ontologies. *The Knowledge Engineering Review*, *17*(2), 129–156. https://doi.org/10.1017/S0269888902000462

Ferreira, A. T., Araújo, A. M., Fernandes, S., & Miguel, I. C. (2017). Gamification in the Workplace: A Systematic Literature Review. *Recent Advances in Information Systems and Technologies. Advances in Intelligent Systems and Computing*, *571*, 283–292. https://doi.org/10.1007/978-3-319-56541-5_29

Friedrich, J., Becker, M., Kramer, F., Wirth, M., & Schneider, M. (2020). Incentive design and gamification for knowledge management. *Journal of Business Research*, *106*, 341–352. https://doi.org/10.1016/j.jbusres.2019.02.009

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, *5*(2), 199–220. https://doi.org/10.1006/knac.1993.1008

Hagedorn, T. J., Smith, B., Krishnamurty, S., & Grosse, I. (2019). Interoperability of disparate engineering domain ontologies using basic formal ontology. *Journal of Engineering Design*, 1–30. https://doi.org/10.1080/09544828.2019.1630805

Hemam, M. (2018). An Extension of the Ontology Web Language with Multi-Viewpoints and Probabilistic Reasoning. *International Journal of Advanced Intelligence Paradigms*, *10*(1), 1. https://doi.org/10.1504/IJAIP.2018.10003857

Hemam, M., & Boufaïda, Z. (2011). MVP-OWL: a multi-viewpoints ontology language for the Semantic Web. *International Journal of Reasoning-Based Intelligent Systems*, *3*(3/4), 147. https://doi.org/10.1504/IJRIS.2011.043539

Heylighen, F. (2016). Stigmergy as a universal coordination mechanism I: Definition and components. *Cognitive Systems Research*, *38*, 4–13. https://doi.org/10.1016/j.cogsys.2015.12.002

Jennings, N. R., Moreau, L., Nicholson, D., Ramchurn, S., Roberts, S., Rodden, T., & Rogers, A. (2014). Human-agent collectives. *Communications of the ACM*, *57*(12), 80–88. https://doi.org/10.1145/2629559

Karacapilidis, N., & Tampakas, V. (2019). On the Exploitation of Collaborative Argumentation Structures for Inducing Reasoning Behavior. *Proceedings of the 18th International Conference on IEEE/Internet*.

Lafleur, M., Terkaj, W., Belkadi, F., Urgo, M., Bernard, A., & Colledani, M. (2016). An Onto-Based Interoperability Framework for the Connection of PLM and Production Capability Tools. *PLM 2016: Product Lifecycle Management for Digital Transformation of Industries*, 134–145. https://doi.org/10.1007/978-3-319-54660-5_13

Liao, Y., Lezoche, M., Panetto, H., & Boudjlida, N. (2016). Semantic annotations for semantic interoperability in a product lifecycle management context. *International Journal of Production Research*, *54*(18), 5534–5553. https://doi.org/10.1080/00207543.2016.1165875

Lundqvist, M. (2007). *Information Demand and Use: Improving Information Flow within Small-scale Business Contexts*.

Oroszi, A., Jung, T., Smirnov, A., Shilov, N., & Kashevnik, A. (2009). Ontology-driven codification for discrete and modular products. *International Journal of Product Development*, *8*(2), 162–177. https://doi.org/10.1504/IJPD.2009.024186

Palmer, C., Urwin, E. N., Young, R. I. M., & Marilungo, E. (2017). A reference ontology approach to support global product-service production. *International Journal of Product Lifecycle Management*, *10*(1), 86. https://doi.org/10.1504/IJPLM.2017.083003

Pew Research Center. (2014). Digital life in 2025. Retrieved 18 August 2020, from http://www.pewinternet.org/files/2014/03/PIP_Report_Future_of_the_Internet_Predictions_031114.pdf

Retelny, D., Bernstein, M. S., & Valentine, M. A. (2017). No Workflow Can Ever Be Enough: How Crowdsourcing Workflows Constrain Complex Work. *Proceedings of the ACM on Human-Computer Interaction*, *1*(2), Article 89. https://doi.org/10.1145/3134724

Scekic, O., Truong, H.-L., & Dustdar, S. (2015). PRINGL – A domain-specific language for incentive management in crowdsourcing. *Computer Networks*, *90*, 14–33. https://doi.org/10.1016/j.comnet.2015.05.019

Shilov, N., Smirnov, A., & Ansari, F. (2020). Ontologies in Smart Manufacturing: Approaches and Research Framework. *2020 26th Conference of Open Innovations Association (FRUCT)*, 408–414. https://doi.org/10.23919/FRUCT48808.2020.9087396

Smirnov, A., Kashevnik, A., Petrov, M., Shilov, N., Schafer, T., & Jung, T. (2019). Competence-Based Language Expert Network for Translation Business Process Management. *2019 25th Conference of Open Innovations Association (FRUCT)*, 279–284. https://doi.org/10.23919/FRUCT48121.2019.8981515

Smirnov, A., Levashova, T., & Shilov, N. (2015). Role-driven context-based decision support: Approach, implementation and lessons learned. In *Communications in Computer and Information Science* (Vol. 553). https://doi.org/10.1007/978-3-319-25840-9_32

Smirnov, A., & Sandkuhl, K. (2015). Context-Oriented Knowledge Management for Decision Support in Business Networks: Modern Requirements and Challenges. *BIR 2015 Workshops, Vol. 1420*.

Smirnov, A., Shilov, N., & Parfenov, V. (2019). Building a Multi-aspect Ontology for Semantic Interoperability in PLM. In *Product Lifecycle Management in the Digital Twin Era. IFIP Advances in Information and Communication Technology* (Vol. 565, pp. 107–115). https://doi.org/10.1007/978-3-030-42250-9_10

Smirnov, A. V., Shilov, N., Oroszi, A., Sinko, M., & Krebs, T. (2018). Changing information management for product-service system engineering: Customer-oriented strategies and lessons learned. *International Journal of Product Lifecycle Management*, *11*(1), 1–18. https://doi.org/10.1504/IJPLM.2018.091647

Staab, S., & Studer, R. (Eds.). (2009). *Handbook on Ontologies*. https://doi.org/10.1007/978-3-540-92673-3

VDMA. German Engineering Federation. (2018). Retrieved from www.vdma.org/en_GB/