This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116870, IEEE Access

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2021.Doi Number

Methodology for Multi-Aspect Ontology Development: Ontology for Decision Support Based on Human-Machine Collective Intelligence

Alexander Smirnov, Member, IEEE, Tatiana Levashova, Andrew Ponomarev, and Nikolay Shilov

St. Petersburg Federal Research Center of the Russian Academy of Sciences, St. Petersburg 199178, Russia

Corresponding author: Tatiana Levashova (e-mail: tatiana.levashova@iias.spb.su).

The research was funded by the Russian Science Foundation under Grant 19-11-00126, except for the competence aspect developed within the State Research, project number 0073-2019-0005.

ABSTRACT The advantage of human-machine collective intelligence for decision support systems is the ability to make better decisions due to the mitigation of human biases in the generation of potential solutions and their evaluation. So far, the potential of human-machine collective intelligence was used only in few decision support systems, however teamwork between humans and machines has not been achieved. This is partly due to the lack of interoperability in these systems. In earlier works, the authors proposed the apparatus of multi-aspect ontologies implying the integration of multiple domain ontologies to provide interoperability between humans and machines and coordinate interrelated processes going on in the systems of the considered type. Such ontologies have proved efficient for systems that require intensive information and knowledge exchange between loosely-related dynamic autonomous domains (e.g., enterprise knowledge management, product lifecycle management, or human-machine collective intelligence systems). However, existing ontology development methodologies fail to recommend a process that would support cross-domain knowledge integration during the multi-aspect ontology development. Moreover, the structure of the multi-aspect ontology imposes some restrictions on the integration approach. The paper proposes such a methodology for the multi-aspect ontology development that incorporates the aspect integration approach at multiple levels. The methodology is applied to develop a multi-aspect ontology for decision support based on human-machine collective intelligence. An example from the “e-tourism” domain demonstrates the applicability of the proposed methodology as well as the usage of the multi-aspect ontology for a human-machine environment aimed at solving real-world problems. The proposed methodology can facilitate the development of ontologies for complex knowledge-based systems that operate with knowledge from multiple loosely-connected domains.

INDEX TERMS Ontology development methodology, multi-aspect ontology, collective intelligence, human-computer teamwork, decision support system.

I. INTRODUCTION

Collective intelligence is an emergent property from synergies among data / information / knowledge, software / hardware, and humans with insight that continuously learns from feedback to produce just-in-time knowledge for better decisions than any of these elements acting alone [1]. A collective intelligence system connecting these three elements into a single interoperable platform, is believed to improve the efficiency of decision support.

Presently, Decision Support Systems (DSSs) that are based on Human-Machine Collective Intelligence (HMCI) are not widespread. Few existing DSSs of this type do not leverage the full potential of HMCI. Basically, they support decision-making by human groups providing them with specially

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116870, IEEE Access

developed software (e.g., [2], [3]), and teamwork between humans and machines has not been achieved so far [4]–[7].

In HMCI-based DSSs, various processes related to different domains (e.g., knowledge management, self-organization, etc.) are ongoing and knowledge from multiple domains is required to support decision-making in such systems. In these systems, humans and machines are supposed to self-organize into teams with a decision support purpose and to interoperate so that they could exchange their views on problems, discuss alternatives, make agreements, etc. The considered systems ensure the interoperability between humans and machines enabling information exchange on the decision support problem, team self-organization problem, and decision-making process [8], [9].

Principal actors distinguished in the HMCI-based DSSs are end-user (decision-maker), participant (human or software service), and service provider [9]. The end-user uses the HMCI-based DSS to get help with decision-making. He/she presents the problem that he/she deals with to the system so that the problem description is visible to the available participants. They self-organize into a team to work on this problem (Fig. 1). The service provider develops software services, integrates them into the DSS, and supports them so that they could act on behalf of the team participants.

Ontologies are an efficient means to support the desired interoperability of the system participants. An ontology for a HMCI-based DSS must meet a set of requirements for humans and software services could interoperate as it is intended in the system. Firstly, such an ontology is required to integrate knowledge of multiple domains that are usually loosely-related. Secondly, since the end-users can deal with various problems from different domains, the ontology is required to be extensible with new knowledge as new problems come. Thirdly, the ontology must provide knowledge enabling the participants to self-organize into teams with a decision support purpose.

The apparatus of multi-aspect ontologies is believed to be a means to meet the requirements above providing a representation of the knowledge of the HMCI-based DSSs, ensuring the interoperability between their components, and

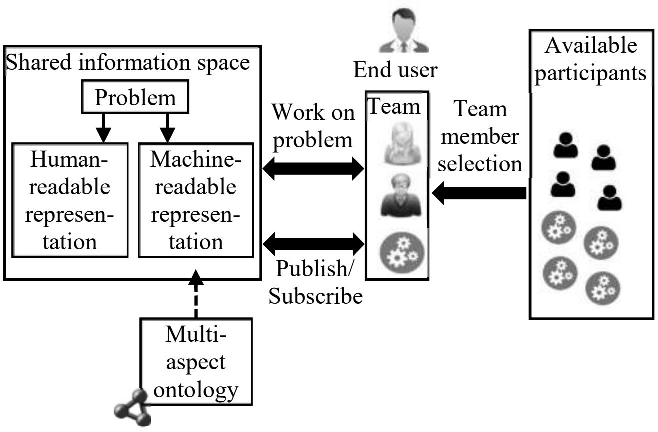

FIGURE 1. Problem processing in HMCI-based DSS.

supporting the coordination of the interrelated processes [10]. It has proved efficient for systems that require intensive information and knowledge exchange between loosely-related dynamic autonomous domains such as enterprise knowledge management [11], [12], product lifecycle management [13], or human-machine collective intelligence systems [14]. A multi-aspect ontology comprises three levels: local, aspect, and global. The local level represents concepts and relationships observed only from one view. Each aspect can be represented by a specific formalism. The aspect level represents concepts and relationships from the local level that are shared by two or more aspects. It defines the formalism of the multi-aspect ontology. The global level is the common part of the multi-aspect ontology represented using the multi-aspect ontology formalism. The concepts represented at this level are related to those of the aspect level.

The intention of the multi-aspect ontologies and their multi-level structure imply the integration of multiple domain ontologies. At that, the ontology structure imposes some restrictions on the integration approach. Though existing ontology development methodologies incorporate activities on ontology integration, they fail to recommend a process that can be used to implement the integration. The above means that developing the multi-aspect ontologies requires a methodology that would incorporate an approach to aspects integration.

The contribution of this work is the methodology for the development of multi-aspect ontologies and a multi-aspect ontology for decision support based on human-machine collective intelligence created following the proposed methodology and applied to an exemplified decision support problem.

The rest of the paper is organized as follows. Related research is outlined in Section II. Section III introduces the methodology for the development of multi-aspect ontologies. The application of the methodology to the development of a multi-aspect ontology for decision support based on HMCI is reported in Section IV. Section V illustrates potentialities of the created ontology when resolving a decision support problem from the "e-tourism" domain by a prototype of the HMCI-based decision support system. Section VI discusses the obtained results. The Conclusion summarizes main research outcomes.

II. RELATED RESEARCH

The presented here research addresses two issues: ontology development methodologies and integration of multiple problem aspects.

A. ONTOLOGY DEVELOPMENT METHODOLOGIES

To date, numerous ontology development methodologies have been published. Since knowledge of multiple domains is involved in the DSSs based on HMCI, methodologies that rely upon knowledge reuse and integration seem to be the most promising. The ontology development methodologies have been analyzed, in particular, regarding these issues (Table I).

One of the first methodologies – "Enterprise" [15] – presents the activities on integrating existing ontologies or ontology reuse as a part of ontology building phase. However, the methodology does not cover the ontology integration part. According to the authors, ontology integration requires an agreement on multiple ontologies that can be shared among multiple user communities, which is a difficult problem.

TABLE I
COMPARISON OF ONTOLOGY DEVELOPMENT METHODOLOGIES

Methodology	Ontology reuse	Ontology integration
Enterprise [15], 1995	-	-
METHONTOLOGY [16], [17], 1997	+	Correspondences between the meta-ontology, the names of terms in the conceptualization, and the names of terms in the ontologies reused (the only methodology considering the implementation of ontology integration)
Protégé [18], [19], 2003	+	-
On-to-Knowledge [20], [21], 2004	+	-
Lifecycles [22], 2013	+	-
NeOn [23], 2015	+	Customization of the general ontology and its integration into the ontology under the development
AMOD [24], 2019	+/-	Integration of ontologies developed during multiple sprints and having homogeneous conceptualizations
Collaborative ontology construction [25], 2021	+	Several possible techniques are referenced as examples

A unified methodology for the development of ontologies – METHONTOLOGY [16], [17] is the only one that embeds integration techniques. The integration activity consists in

– choosing an existing meta-ontology or implementing a new meta-ontology;

– searching for ontologies providing definitions of terms whose semantic and implementation are coherent with the terms identified in the conceptualization;

– producing an integration document that provides correspondences between the meta-ontology, the names of terms in the conceptualization, and the names of terms in the ontologies reused;

– implementation using an environment that supports the meta-ontology and ontologies selected for the integration.

The Protégé methodology [18], [19] suggests several reasons why reuse and integration of existing ontologies are important, but it does not provide any views on the integration.

The On-to-Knowledge methodology [20], [21] considers ontology integration at the ontology refinement phase, but the methodology does not propose any substeps dealing with this specific problem. However, ontology reuse is still assumed.

The NeOn methodology [23], which is a scenario-based methodology that supports the construction of ontologies and ontology networks by reusing available knowledge resources (ontologies, non-ontological resources, and ontology design patterns), proposes a set of activities for different scenarios dealing with the integration of reused knowledge. The integration of ontological resources relies upon a general ontology that is selected from the existing ontologies and reused. A general ontology that best fits the requirements to the ontology to be developed is selected, customized, and integrated into the ontology under development.

In the methodology that focuses on the ontology evaluation across its lifecycle [22], the ontology development phase covers both new ontology development and ontology reuse, but because the focus of the methodology is the ontology evaluation, the ontology integration techniques are out of the methodology scope.

The Agile Methodology for Ontology Development (AMOD), which integrates agile principles and practices in the ontology development [24], ontology integration is used to integrate ontologies developed during multiple sprints. Basically, it is an integration of conceptualizations, which does not address the problem of resolving heterogeneity of the representations.

The improved methodology for collaborative construction of reusable, localized, and shareable ontologies proposed in the domain of Interlocking Institutional Worlds aims to adapt the best practices from the ontology development methodologies [25]. The methodology assumes ontology integration to include ontologies of related domains or sub-domains into the developed ontology. Nevertheless, no specific integration techniques are proposed (several techniques are referenced as examples). A distinctive feature of the methodology is that it assumes using an existing upper ontology, in particular Unified Foundational Ontology (UFO) [26], for modeling the created ontology. Based on this fact, it can be assumed that the upper level ontology supports the ontology integration.

In the TOVE methodology [27], which provides a logic-based formal approach that transforms informal scenarios into a computable model expressed in first-order logic, ontology reuse and integration are out of the methodology scope. As a result, it is not included into the table.

The analysis of the ontology development methodologies has shown that almost all the methodologies take into account ontology reuse and integration. The methodologies that provide some suggestions on the integration approach propose usage of a top-level ontology or a shared ontology to support the integration. While an agreement on a top-level ontology is not always achievable, especially when it comes to multiple domains, a shared ontology seems to be a practically acceptable approach.

In addition, the analysis above enabled us to reveal an ontology development pattern that the methodologies follow: requirements specification, creation of conceptualization, conceptualization formalization, ontology implementation, and ontology evaluation. An important part of the methodologies is competency questions. Mainly, they are used for the specification of ontology requirements and ontology evaluation.

Two reasons cause the need to develop a methodology for the multi-aspect ontology development. The first one is that multiple domains involved in the DSSs based on HMCI make it not always possible to use the same shared ontology for various decision support problems. The terminology and notations used in various processes taking place in a HMCI-based DSS may differ since these processes are aimed at solving tasks of different nature that require different techniques [28], [29]. The second reason is that the multi-aspect ontologies rely upon a certain structure that implies multi-level aspect integration. Thus, the present research suggests the shared ontology to be a part of the multi-aspect ontology and considers it as one of the results of the multi-level aspect integration.

B. INTEGRATION OF MULTIPLE ASPECTS

Approaches aiming at a multi-aspect problem representation and therefore forced to integrate multiple problem aspects have been analyzed according to the criteria below.

– Support of heterogeneous formalisms. Does the ontology support integration of heterogeneous knowledge at that preserving its original terminologies and notations?

– Maintenance of changes in aspects. Does the ontology take into account the changes occurring in the aspects and in what way?

– Automatic aspect integration. Does the ontology provide mechanisms for automatic knowledge integration?

– Support of loosely connected aspects. How tightly are interrelated domains the knowledge from that the ontology integrates?

– Openness for new knowledge. Does the ontology enable one to supplement it with new knowledge?

– Integration mechanism. What kind of knowledge integration does the ontology support?

One of the analyzed approaches is model-driven interoperability framework. This framework is aimed at supporting relationships between products and manufacturing equipment, a "connection framework" describes relationships between different ontologies of products maintained in a Product Lifecycle Management system and different ontologies of manufacturing capabilities managed in the Manufacturing Process Management system [30]. It doesn't support direct translation of information from one specific ontology to another ("aspect" ontologies). The translation between the source ontology and a common shared ontology is followed by the translation between the common ontology and the target ontology. Even though such ontology alignment language as EDOAL [31] have a high level of expressiveness, different formalisms may not necessarily support features of other formalisms what will cause the loss of information and knowledge. Changes in aspect ontologies do not necessarily require changes in the common ontology, however any change in an aspect ontology would require its re-alignment with the common ontology.

Knowledge integration to represent multiple aspects of the same knowledge is a focus of multilingual ontologies [32]–[34]. They integrate domain knowledge described by different languages. Their goal is to resolve terminological problems caused by the absence of exact equivalents of terms in different languages. Multilingual ontologies are created in a modular way so that they have fragments associated with specific languages linked with special relationships. Such ontologies have been intensively developing. Multiple approaches propose different frameworks to the construction of these ontologies including usage of shared ontologies [35] or thesauruses [36] and semi-automatic [37] or automatic [38] ontology creation. As a rule, multi-lingual ontologies support augmentation with new knowledge or even can be a result of new knowledge discovery [39]. The multi-lingual ontologies can be thought of as a solution for the multi-aspect problem representation if they consider the fragments as aspects. However, one of the main specifics of the multi-aspect integration is that different processes use different formalisms for knowledge representation, which is not supported by the multilingual ontologies.

Granular ontologies offer another way to integrate multiple aspects [40], [41]. They propose granular perspectives for a specific ontological commitment. The granules can be viewed as pieces of knowledge about object, process, or sub-domain in a certain perspective. The ontology commitment is invariable, which makes it difficult to maintain changes in the granules and introduce new knowledge. Depending on the contiguity of the described objects, processes, or sub-domains the granules can be closely connected, loosely connected or not connected at all. Despite the advantages of the granular ontologies as consistent terminology among the granules of the same knowledge and a granule hierarchy based on different levels of detailing where each level corresponds to a perspective, they cannot solve the problems of having different notations for different granules and resolving ambiguities as multilingual ontologies do.

Viewing a domain from different viewpoints has resulted in appearance of Multi-Viewpoints Ontology (MVpOnt) where each viewpoint corresponds to the knowledge representation model, which is useful for a particular task, process or a group of people, which co-exist in a common information environment and share some information and knowledge [42], [43]. Thus, the viewpoints of MVpOnt are aimed at describing the same object or process though in different perspectives, and therefore are semantically close. Ontology elements (classes and class properties) are divided into two groups: local – observed only from one viewpoint, and global –

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116870, IEEE Access

observed from two or more viewpoints and described in a shared level. "Bridge rules" relate concepts from different viewpoints. In case of changes in viewpoints or adding new viewpoints the shared level can be adjusted (this process is not automated), but this is required only if global elements are affected. This approach seems to be the most suitable since it does not only support resolving terminological issues but also makes it possible to preserve original formalisms used in existing ontologies.

The analysis of the multi-aspect problem representation approaches has shown that in these approaches the integration of knowledge represented by heterogeneous formalisms either is not supported or leads to some loss of expressiveness of the representations. The multi-viewpoint approach seeming the most suitable concentrates on the integration of the closely related viewpoints on a problem. This approach can be taken as the basis and adapted to loosely connected domains.

A comparison of the described approaches to multi-aspect integration with the mechanism of the multi-aspect ontologies is presented in the Discussion.

III. METHODOLOGY FOR DEVELOPMENT OF MULTI-ASPECT ONTOLOGY

The proposed methodology of the multi-aspect ontology development adopts the pattern that the analyzed methodologies follow. The methodology divides the ontology building process into four stages (Fig. 2):

The first stage aims at producing ontology requirements specification, the identification of the purpose and scope of the ontology, and the identification of aspects to be included in the multi-aspect ontology. The methodology does not impose special demands on requirements specification methods. Nevertheless, competency questions that most of the ontology development methodologies recommend can be used for this objective. Based on the requirements specification the purpose and scope of the ontology are identified, they in turn provide ideas about the kinds of aspects.

The second stage focuses on the development of aspect ontologies. The stage starts with the requirements specification for each aspect. Then, two scenarios of the aspect ontologies development can be used: development from scratch or ontology reuse.

Ontology reuse is considered to be of a higher priority. The reused ontologies are adjusted to the aspect ontologies requirements without changing the original formalisms of those ontologies. The adjusted ontologies are evaluated. In respect that the developers of the reused ontologies evaluated them, the evaluation concerns checking the consistency of the representations for the concepts and properties that have been changed while the adjustment. These representations must correspond to the representations used in the source ontologies. Here, consistency of the representations means correct spelling of names for the concepts and properties, consistency of their grammatical forms, lack of redundancy, etc. In detail, the Protégé methodology [19] describes this kind of evaluation referred there as verification.

If no ontologies for reuse have been found, the aspect ontology is built from scratch following the revealed ontology

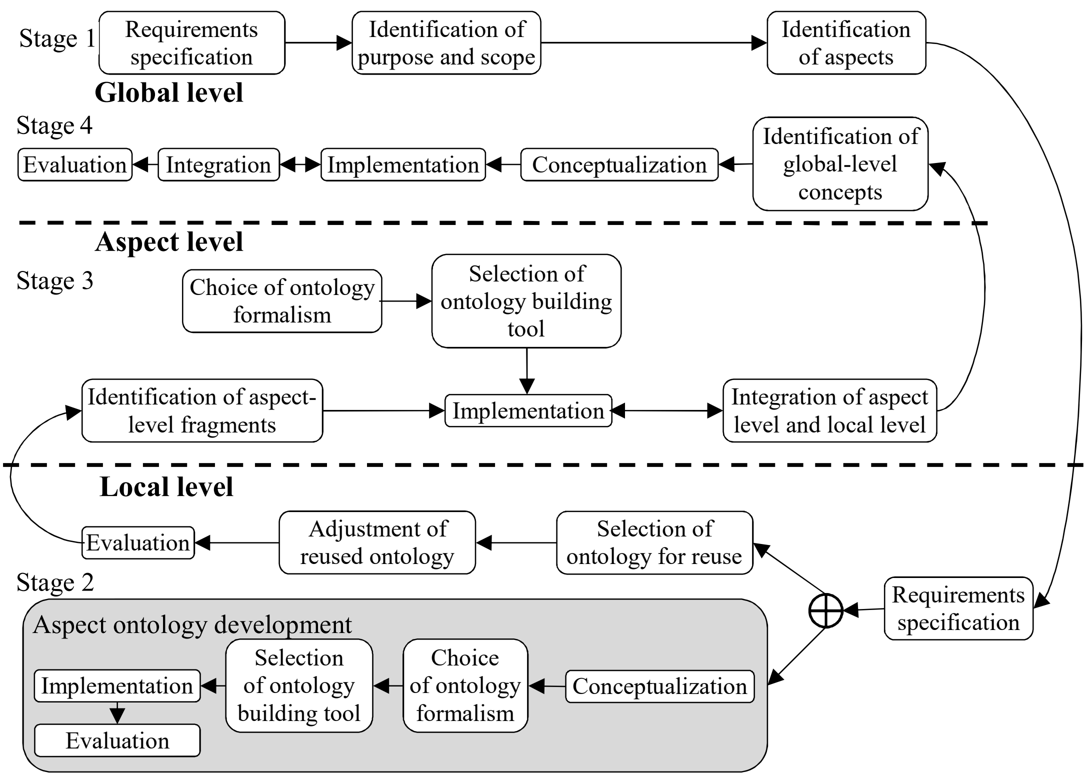

FIGURE 2. **Multi-aspect ontology development.**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116870, IEEE Access

development pattern: requirements specification, conceptualization, formalization, implementation, and evaluation. The methodology of the multi-aspect ontology development addresses the issues of aspect ontology formalization and implementation jointly since they can be supported by various existing ontology building tools. They support both the aspect ontology representation in a chosen formalism or representation language and the ontology encoding, i.e. its implementation. The resulting aspect ontologies are verified for the representation consistency and evaluated for the logical consistency. The logical consistency can be checked by an ontology reasoner usually integrated into ontology building tools.

The third stage targets representing the aspect level of the ontology. At this stage, fragments of the aspect ontologies are identified that represent concepts of the aspect level and relationships between them. As the aspect-level concepts, we capture concepts for which there exist semantic mappings to the concepts of one or more aspect ontologies.

The ontology fragments are represented in the same formalism (hereinafter referred to as the multi-aspect ontology formalism) that does not depend on the formalisms of the (local) aspect ontologies. The multi-aspect ontology formalism and an ontology building tool to represent the fragments can be chosen before, during, or after the development of the aspect ontologies depending on the developers' goals. If the local level represents many aspect ontologies implemented using the same formalism, sometimes it makes sense to represent the aspect level using the formalism of those ontologies. However, this recommendation is not mandatory.

Evaluation of the aspect-level fragments consists in checking whether the conceptualization formalized by a fragment complies with the conceptualization of the aspect ontology the fragment has been extracted from.

The final activities at the third stage concern integration of the aspect and local levels. They are integrated via unambiguous alignment of the aspect-level fragments and the aspect ontologies. The alignment relationships are formalized using aspect ontology formalisms.

The fourth stage is intended for the development of the global level of the ontology. For this purpose, the concepts common to several aspect-level fragments are identified. Concepts of the global level are selected among the common concepts and then a conceptualization of the global level is created. This conceptualization is formalized and implemented using an ontology building tool (usually, the tool chosen at stage 3).

The global level and the aspect level are integrated through bridging rules that are specified between the concepts of these two levels. These rules are formalized by means of the multi-aspect ontology formalism. Bi-directional arrows in Fig. 2 between the implementation and integration blocks indicate that the activities on the ontology implementation and ontology integration are interrelated. In particular, one can integrate knowledge when creating a conceptualization, and then implement this conceptualization, or one can introduce (integrate) some knowledge into the implemented ontology and therefore to change the original conceptualization.

The evaluation of the multi-aspect ontology finalizes the fourth stage. The evaluation implies checking the global level representation consistency, checking for the lack of redundant relationships at this level, and checking the logical consistency of the aspect level and the global level. At the fourth stage, the logical consistency is checked via passing facts between the aspect and global levels using the bridging rules.

IV. DEVELOPMENT OF MULTI-ASPECT ONTOLOGY FOR DECISION SUPPORT BASED ON HUMAN-MACHINE COLLECTIVE INTELLIGENCE

The development of the multi-aspect ontology for decision support based on human-machine collective intelligence follows the stages proposed by the methodology introduced above. The ontology development starts and finishes with the design of the global level, where at stage 1, ontology aspects are identified, and at stage 4, the conceptualization of this level is created. Between stages 1 and 4, at stage 2 aspect ontologies are developed, and at stage 3 the aspect level of the multi-aspect ontology is built.

STAGE 1. GLOBAL LEVEL

Specification of requirements to the multi-aspect ontology.

The purpose and scope of the multi-aspect ontology and competency questions form the basis to produce ontology requirements specification. This specification comprises a set of initial concepts considered relevant to the conceptualization of the modeled domain. With reference to the global level, these concepts correspond to the kinds of aspects viewed in the ontology.

The purpose of the multi-aspect ontology is to provide interoperability for *components* of a *human-machine environment* that provides *collective intelligence*.

The ontology scope is the processes related to solving the user *task* as a *decision support problem* by a *human-machine environment*.

Relevant concepts revealed from the ontology purpose and scope are *human-machine environment*, *component*, *collective intelligence*, *decision support*, *problem*, *task*.

Competency questions:

– What entities are participants of a human-machine environment?

– What knowledge should the ontology represent to provide the organization of a human-machine environment?

– What knowledge should the ontology represent to provide decision support?

– What knowledge should the ontology represent to be able to treat the user task?

Answers to the competency questions. *Software services* and *humans* are *participants* of a *human-machine environment*. They *self-organize* based on their *competencies* to solve the user task as a *decision support* problem. The human-machine environment provides the user with a task solution that is the result of *decision making*. The scope of the user task is restricted by the *application domain*.

Relevant concepts revealed from the competency questions are *human-machine environment, participant*, *software service*, *human*, *self-organization*, *decision support, decision making, competency, application domain.*

Identification of aspects to be included in the multi-aspect ontology.

Ontology aspects are chosen based on the revision of the set of initial concepts. The revision concerns the identification of possible synonyms, subconcepts, irrelevant concepts, or lacking concepts. The revision showed that *component* and *participant* are synonyms regarding the human-machine environment. These concepts are decided to be referred to as *participant*. At the same time, *participant* is a subconcept for the concept of *human-machine environment,* and therefore there is no need to represent *participant* as a separate aspect. The concepts of *software service* and *human* are kinds of *participants* and consequently are not represented as aspects. The *user task* in terms of decision support is considered as *problem*, hence *problem* and *task* are synonyms, and the term *problem* is chosen to name these concepts. The concept *problem* belongs to the *decision support* domain and is used in conjunction with knowledge of the *application domains* that supply the input data for this *problem*. *Decision making* is the set of activities supported by the decision support system and therefore is considered within the *decision support* domain. *Collective intelligence* is the product of activities of the human-machine environment participants and cannot be considered as an aspect.

Summing up, the following aspects that should be introduced into the multi-aspect ontology are identified: *decision support*, *application domain*, *human-machine environment*, *self-organization*, *competency*, and *application domain*

STAGE 2. LOCAL LEVEL

At stage 2, ontologies for the aspects identified above are developed. This can be done based on any ontology development methodology since the aspects are generally independent (i.e., they can be implemented using different formalisms and representation languages). As it is said above, ontology reuse is preferable.

In the paper, ontologies for the aspects of *decision support, human-machine environment*, *self-organization*, and *competency* are developed from scratch. The application domain ontology is reused. The present paper is limited to presenting the aspect conceptualizations.

The creation of conceptualizations starts with the identification of relevant aspect concepts. Relevant concepts for the aspects of *human-machine environment*, *self-organization*, and *competency*, are identified in the same way as for the global level (ontology purpose, scope, competency questions). Relevant concepts for the *decision support* aspect are identified based on the definitions for the ontology purpose and scope, and an analysis of decision making methodologies. The formulation of the problem to the DSS by the user provides relevant concepts for the *application domain* aspect.

In this section, the conceptualizations for the aspect ontologies of *decision support, human-machine environment*, *self-organization*, and *competency* are described. These ontologies are independent of application domains. Usage of these ontologies in a real example is reported in Section V, where an ontology for the application domain is presented.

DECISION SUPPORT.

The purpose of the decision support aspect is the support of the decision maker with decisions. The scope of this aspect is the support of a user with decisions for the problem with that this user deals in the current situation (context). Relevant concepts revealed from the ontology purpose and scope are *decision support*, *decision*, *decision maker*, *user*, *problem*, *context*.

A set of candidate concepts has been obtained as a result of the analysis of decision-making methodologies [44]–[55]. At the conceptualization stage, this set was revised. A decision support ontology for collaborative decision making in engineering design [56], [57] and a Sample Decision Ontology [58] were used for reference when revisioning the concepts.

Table II summarizes the names for the concepts introduced to the ontology and shows the correspondences between these names and references to these concepts in the decision-making methodologies.

The conceptualization for the decision support aspect is given in Fig. 3. It comprises the following concepts.

Decision support system – an information system that supports decision-making activities.

Context – information that can be used to characterize the decision situation; context represents reasons, facts, contradictions, and other situation-related information.

Problem – an issue that has to be resolved by finding an answer to it or taking some actions.

User – one who uses the decision support system as a decision support tool.

Decision maker – one who makes the final choice among the alternatives.

Role – decision support and decision-making activities required or expected of the users and decision-making participants within the role.

Criterion – a rule or standard by which alternatives can be ranked based on the decision-maker preferences.

Preference – objective function and the desired value of the objective function.

Alternative – optional problem solutions or courses of action.

Decision – an agreement to adopt an alternative to resolve the problem.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116870, IEEE Access

TABLE II
ONTOLOGY CONCEPTS

Methodology concept	Ontology concept	Definition
Alternative, Assumption, Option, Possible action, Possibility	Alternative	Optional problem solutions or courses of action
Controversions, Information, Facts, Reality, Situation	Context	Any information that can be used to characterize a situation [59]
Criteria, Benefits, Effectiveness, Immediacy, Long term, Optimality, Practicability, Priority, Pros and Cons, Risks, Satisfactoriness, Weight	Criterion	A rule or standard by which alternatives can be ranked based on the decision-maker preferences
Conclusion, Decision, Solution	Decision	Agreement to adopt an alternative to resolve the problem
Role	Decision maker	A participant of the human-machine community that makes the final choice among the alternatives
Analysis, Checking, Assessment, Evaluation, Justification, Learn from, Reconsideration, Reflect, Review, Verification	Evaluation	Judgment how much the object being evaluated (context, participant, decision, result) addressed what is expected
Resource	Participant	Human or software service taking part in decision making
Preference	Preference	The objective function and the desired value of the objective function
Dilemma, Goal, Problem	Problem	A question that has to be resolved by finding an answer to it or taking some actions
Consequences, Outcome, Result	Result	Events due to the decision
Role	Role	Decision-making activities required or expected of a participant
-	Activity	Participant actions related to the achievement of any goal while decision-making (e.g., information gathering, context analysis, development of alternatives, etc.)
-	Decision making	The process that is used to make a decision being solution to the problem

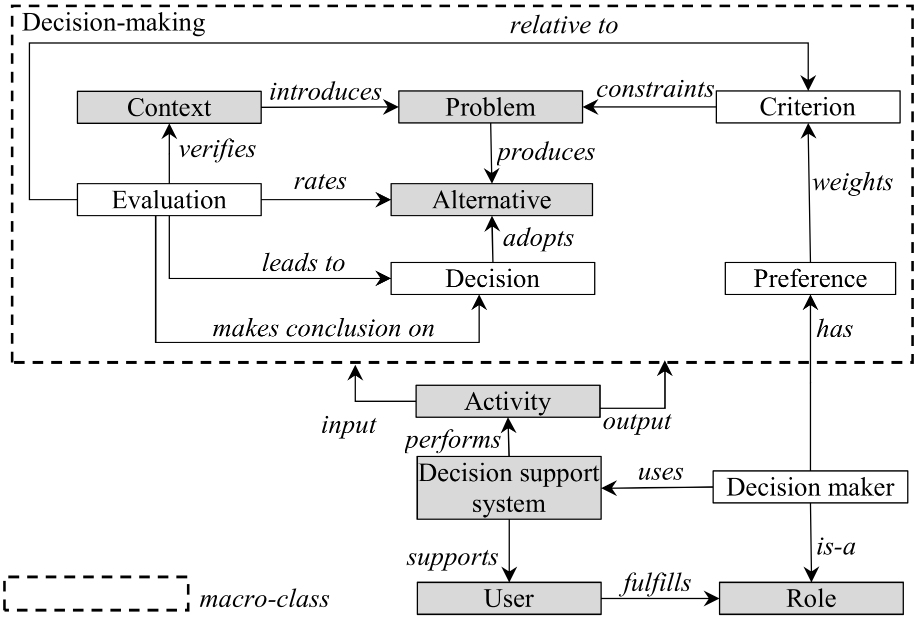

FIGURE 3. Conceptualization for decision support aspect.

Evaluation – a judgment of how much the object being evaluated (the context, the human-machine environment, the decision) addressed what is expected.

Activity – actions related to the achievement of any goal while decision support (e.g., information gathering, context analysis, development of alternatives, etc.).

Decision making – a process that is used to make a decision being solution to the problem.

The concepts are in the following relationships.

An appearance of difficulty or contradiction in a situation (*context*), which is required to be resolved, causes *introducing* a *problem*. The context is verified by the *decision support system*. Typically, there exist a set of *alternatives* each of which provides a solution to the problem (in the conceptualization it is represented as problem *produces* alternatives). These alternatives are evaluated via their *rating* based on a set of *criteria* that one fulfilling the *role* of *decision maker* selected by *weighing*. The *evaluation* of the alternatives *leads to* a *decision* that is the preferred alternative *adopted* as the alternative that satisfies the *criteria* most closely. The *decision* is *analyzed to make conclusion* whether it has led to the problem

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116870, IEEE Access

resolution. The *activities* on the problem identification, development of alternatives, evaluation, and making a decision are *fulfilled* by the *user* and the *decision-making* participants in accordance with their *roles*. *Outputs* of some activities serve as *inputs* for others.

HUMAN-MACHINE ENVIRONMENT

The purpose of the human-machine environment aspect: this aspect to be used to resolve the *problem* at hand through *teamwork* between *software services* and *experts*. The aspect scope is the *problem* formulated by the *user*.

Competency questions:

– What entities are participants of a human-machine environment?

– How is the problem resolving process represented?

– How do the participants of the human-machine environment relate to the problem?

– How does the human-machine environment take into account the current situation when problem resolving?

Answers to the competency questions. *Software services* and *humans* are the *participants* of a *human-machine environment*. A *process model* represents the subprocesses of the *problem resolving process*. Participants organize a *team* using a *process model*. This model is described by a *meta-model* applying *process patterns*. The team implements the *process model* for a specific *problem*. Information about the current situation comprises the *context*, the *information on the problem* from the *user*, and the *information on the problem* produced by the team in the course of its *activities* on *problem-resolving*.

Relevant concepts revealed from the purpose, scope, and competency questions are *human-machine environment*, *user*, *participant*, *software service*, *expert*, *human*, *teamwork*, *team*, *problem*, *problem-resolving process*, *activity*, *process model*, *meta-model*, *process pattern*, *context*, *information on the problem*.

Then, the domain experts revised the list of the revealed concepts and suggested the following revisions:

– human and expert are synonymous; it was decided to use the *expert* concept in the conceptualization;

– teamwork supposes a set of activities that individuals undertake to achieve a common goal. In the human-machine environment, this goal is to resolve the problem. Here teamwork means problem resolving by the team; the experts decided to exclude teamwork from the conceptualization and retain the concepts of *problem resolving* and *team*;

– problem-resolving (or problem-resolving process) is represented by the process model. The concept of *process model* was included in the conceptualization;

– the process model implies the activity concept; consequently, activity is excluded from the conceptualization.

Besides, the domain experts decided to extend the aspect ontology with the classification of software services, and concepts for the specification of *software service providers*, *agents of external sources of data/knowledge*, and *expert agents*. In addition, they proposed to introduce the concept of *artifact* to represent any results created by the participants as an outcome of their activities.

The conceptualization for the aspect of the human-machine environment is given in Fig. 4. It comprises the following concepts.

Human-machine environment – a community of software services and machines the interactions of which leads to the emergence of collective intelligence

User – one who formulates the problem to the human-machine environment, monitors the progress of its resolving, makes adjustments to the resolving process, and makes a final decision based on the information provided by the environment.

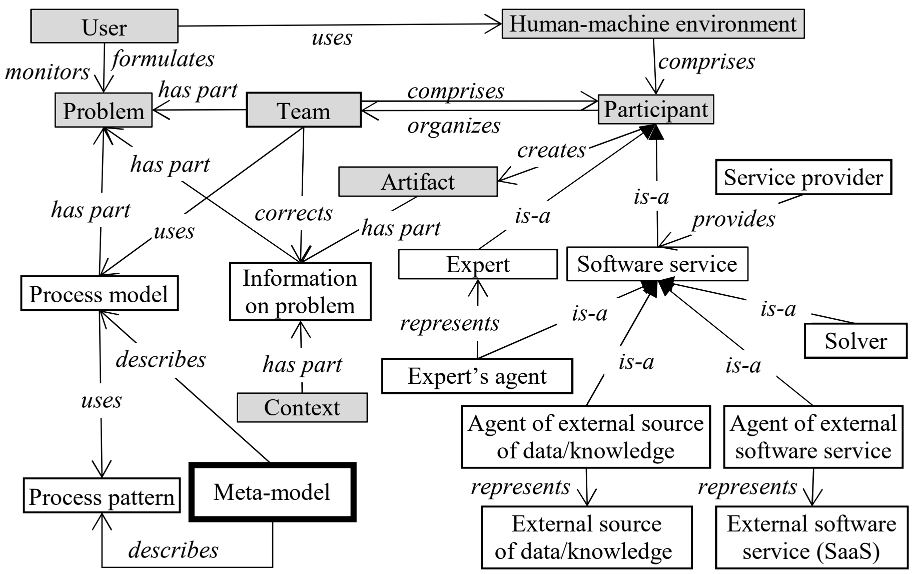

FIGURE 4. Conceptualization for human-machine environment aspect.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116870, IEEE Access

Problem – a matter which is difficult to solve or settle, a doubtful case, or a complex task involving doubt and uncertainty.

Participant – a software service or human taking part in problem resolving.

Software service – software provided as service.

Service provider – a person or a company proving the services to the subscribers.

Expert – human as a participant of the human-machine environment.

Expert's agent – a software service acting on behalf of the expert.

Team – a set of participants implementing the process model (the model of the problem resolving process is meant) for a specific problem.

Artifact – a result created by a participant as an outcome of the participant's activity.

Agent of an external source of data/knowledge – software service as a participant that provides access to an external source of data/knowledge

External source of data/knowledge – a ready-to-use source of data or knowledge (e.g., a database or a service) that is hosted, supported, and maintained by a source provider and to which the environment has access under specified conditions.

Agent of external software service – software service as a participant that provides access to an external software service.

External software service – ready-to-use software service that is hosted, supported, and maintained by a service provider and to which the environment has access under specified conditions.

Solver – intelligent software that synthesizes information and knowledge to achieve a solution for subproblems appeared while the participant activities on problem resolving.

Process model – a description of the problem-resolving subprocesses (e.g., data collection, situation assessment, investigation, etc.) created using the meta-model.

Meta-model – a set of elements for the creation of a given process model.

Process pattern – a solution for typical tasks of coordination and organization of participants of the human-machine environment to resolve the problem.

Information on problem – context, information on the problem from the user, and information on the problem produced by the team in the course of its activities on problem resolving.

Context – any information that characterizes the situation of an entity where an entity can be a place, a participant of the human-machine environment, or the user.

The relationships between the concepts are as follows:

Participants of the human-machine environment *are* software services and experts. The user *uses* the environment to resolve the *problem* that he/she has faced. The user *formulates* the problem and *monitors* the progress of its solution by the participants. The participants *organize* a team (team *comprises* participants) to resolve the problem jointly. The team is organized *using* a process model that is *described* by the meta-model *using* process patterns. In the problem-resolving process, the participants *create* artifacts; these artifacts become *part of* the information on the problem. In addition to the artifacts, the information on the problem includes (*has part*) context. Experts are *represented* by their agents. Service providers *provide* software services. If data/knowledge that is external to the human-machine environment is required then the source of this data/knowledge is accessed under the conditions agreed with the source provider. In this case, the accessed source is *represented* in the human-machine environment by the agent of an external source of data/knowledge. If the software that is external to the human-machine environment is required (e.g., computational services) then the software service is accessed under the conditions agreed with the service provider. In this case, the accessed service is *represented* in the human-machine environment by the agent of external software service.

SELF-ORGANIZATION

The aspect purpose: the aspect of self-organization to be used to enable independent *entities* to *self-organize* into a *team* for their joint activities aimed at achieving a common *goal*. The aspect scope is *solving* a *task*.

Competency questions:

– What is the *motivation* of independent *entities* to participate in *self-organization*?

– How is the *motivation* related to *remuneration*?

– How is the *remuneration* distributed between the *team participants*?

– What is the *purpose* of self-organization?

– How is the *process* of self-organization described?

– How are the functions of the team participants described in the self-organization process?

Answers to the competency questions. Independent *entities* are motivated to participate in the *self-organization* because they expect some *remuneration* for their activities in a self-organized *team*. An entity gets some *remuneration* that *corresponds* to the entity's *motivation*. The *purpose* of the self-organization is *task solving*. The entities apply a *self-organization model* in the *process of their self-organization*. This model determines *functions* that the entities perform when self-organizing.

The revealed relevant concepts are *self-organization*, *entities*, *team*, *task solving*, *motivation*, *remuneration*, *purpose*, *participant*, *self-organization process*, *self-organization model*, *function*.

Revisions made by the domain experts are as follows. Similar to how it was decided for the process concept in the human-machine environment aspect, the concept of *self-organization model* instead of self-organization process to be used in the conceptualization of the self-organization aspect. The concept of *self-organization method* to be used instead of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116870, IEEE Access

function because such a method specifies the functions that the entities perform when self-organizing.

Moreover, the experts pointed out that any task progresses through a series of states from formulation until solution, i.e. a task has the *lifecycle*. Each state requires different activities from the team participants. These activities the participants carry out within their *roles*. The roles require participants to have appropriate *competencies*. The experts proposed to model in the conceptualization the idea of relating the participant activities according to the roles they fulfill with the task lifecycle states.

The conceptualization for this aspect is given in Fig. 5. It comprises the following concepts.

Entity – some identity capable to self-organization.

Self-organized team – a team that has the autonomy to choose how best to accomplish their work.

Participant – a member of the self-organized team that takes part in the task lifecycle in accordance with the role that this participant fulfills.

Lifecycle – a course of changes through which a task passes starting from the task formulation until a solution is found.

Role – a set of activities assigned to a specific role that is performed within the task lifecycle.

Motivation – a reason or reasons for acting or behaving in a particular way.

Remuneration – any kind of compensation for the performing activities supposed by a role.

Self-organization – a process resulting in some form of order due to interactions between independent entities.

Purpose – a goal of self-organization that is to solve the task.

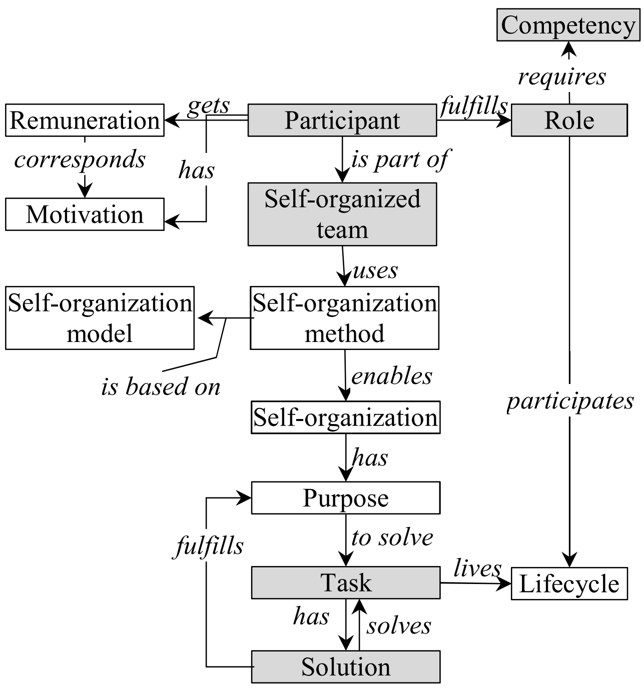

FIGURE 5. Conceptualization for self-organization aspect.

Self-organization model – a way to reach such an agreement between the participants without interference from outside, which enables the participants to solve the task.

Self-organization method – a set of functions that provides self-organization in accordance with the self-organization model.

Task – an issue that the self-organized team is expected to solve.

Solution – the result obtained as the purpose achievement.

Relationships between the concepts are as follows. Independent entities become participants of a self-organized team as the result of their self-organization. The participants are *part of* this team. The self-organization of the participants is *enabled* by a self-organization method, which *is based on* a self-organization model. The purpose of self-organization (self-organization *has* a purpose) is *to solve* a task. The participants *get* remuneration for the participation in self-organizing and task solving. The remuneration *corresponds* to the participant's motivation. The task *lives* its lifecycle. The participants *participate* in the task lifecycle in accordance with the roles that they *fulfill*. A task *has* a solution. The solution *solves* the task and *fulfills* the purpose.

COMPETENCY

The purpose of the competency aspect: the *competency* aspect is intended to represent knowledge that can be used to identify *entities* capable to solve a *task* and to identify *tasks* that an *entity* can solve. The aspect scope is *solving* a *task*.

Competency questions:

– What kinds of *tasks* is the *entity* capable to solve?
– What kinds of *tasks* determine *competency* X?
– Does *task* Y require a single kind of *competence* for its solution?
– Which competencies are varieties or *alternatives* of *competency* X?

Answers to the competency questions. An *entity* is capable of solving a *task* for which this entity has appropriate *competency*. *Competency* X determines the kinds of *tasks* that an *entity* with this *competency* must be able to solve. One or more *competencies* can be required to solve *task* Y. *Competency* X may have *alternative competencies* that include the capabilities to solve the given *task* among capabilities to solve other kinds of tasks.

The list of the relevant concepts comprises *entity*, *competency*, *task*, *alternative competency*.

The conceptualization for the competency aspect is given in Fig. 6. The following concepts make up this conceptualization.

Entity – something or someone possessing some knowledge, skill, or ability to solve the task.

Competency – capability to solve a certain class of tasks.

Task – an issue that needs to be resolved by an entity based on the competencies of this entity.

The concepts are in the following relationships. Entities *have* competencies. The competencies *determine* the kinds of tasks that the entities having these competencies The

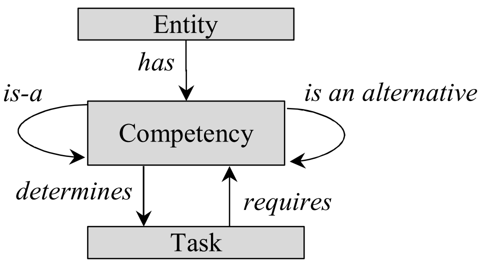

FIGURE 6. Conceptualization for competency aspect.

competencies *determine* the kinds of tasks that the entities having these competencies have to be able to solve. One or more competencies are *required* to solve a task. The competencies are organized in a taxonomy (*is-a* relationship) and have *alternative* competencies.

STAGE 3. ASPECT LEVEL

The shadowed blocks in Fig. 3 – Fig. 6 represent concepts that are chosen as aspect-level concepts. Table III summarizes these concepts.

TABLE III
CONCEPTS OF THE ASPECT LEVEL

Local-level aspect	Aspect-level concept
Decision support	decision support system, user, role, problem, context, alternative, activity
Human-machine environment	human-machine environment, user, participant, problem, artifact, context, team
Self-organization	self-organized team, participant, role, task, solution, competency
Competency	entity, competency, task

Fig. 7 shows fragments of the aspect ontologies, in which the aspect level concepts and the relationships between them are represented. According to the methodology, concepts of the (local) aspect ontologies and concepts of the aspect level are aligned.

The OWL ontology description language [60] is chosen for the aspect level formalization since it is the most widely used up-to-date means supported by the World Wide Web Consortium (W3C). Most of the existing ontologies have representations in this language. The Protégé ontology editor and building framework [61] is used to implement the aspect level.

The Multi-Viewpoint OWL (MVP OWL) extension [42], [62] has been selected as the basis for the development of the formalism for multi-aspect ontology representation since it is mostly oriented to the representation of a single ontology including different points of view rather than alignment and matching of ontologies.

STAGE 4. GLOBAL LEVEL.

The concepts chosen as the global level concepts and the correspondences between these concepts and aspect level concepts are presented in Table IV. In the table, the aspects are referred to as follows: DS – decision support aspect, HME – human-machine environment aspect, SO – self-organization aspect, C – competency aspect. The global level of the multi-aspect otology is presented in Fig. 8.

Bridging rules formalize the correspondences between the global-level concepts and aspect-level concepts. According to the MVP OWL, all OWL classes and properties within the ontology are divided into global (visible from two or more aspects) and local (visible only from one aspect):

C^G – the set of global classes;

P^G – the set of global properties.

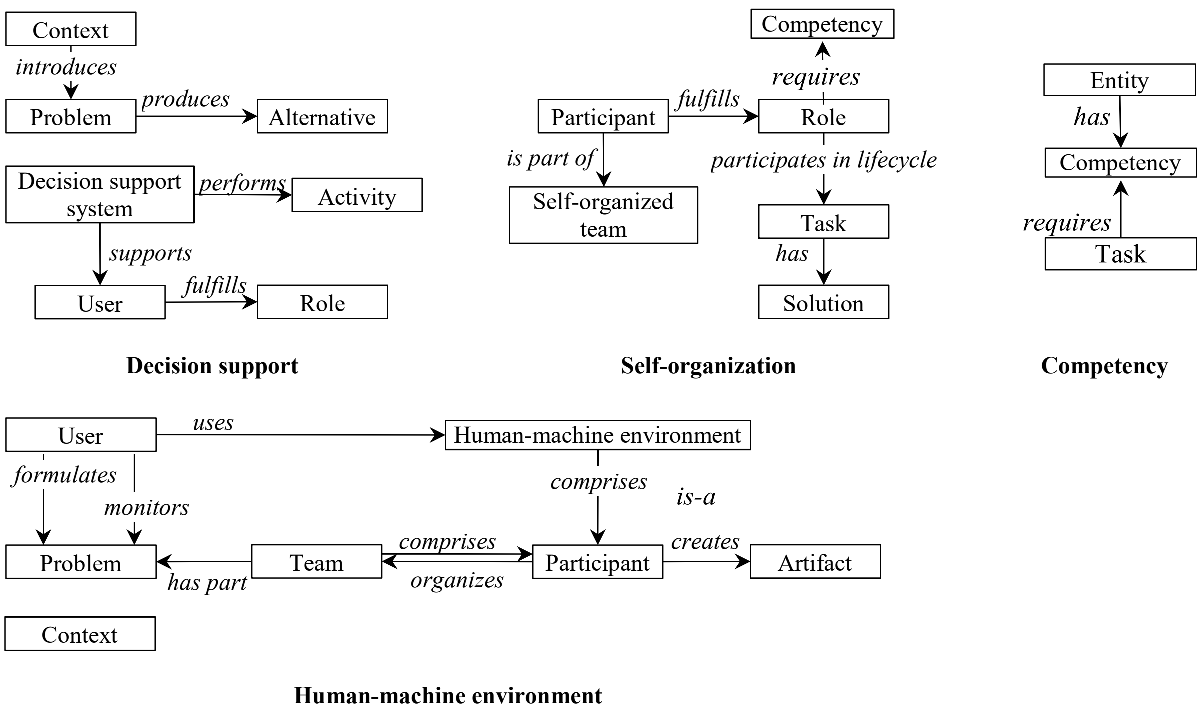

FIGURE 7. Aspect level: fragments of aspect ontologies

TABLE IV
CORRESPONDENCES BETWEEN GLOBAL AND ASPECT LEVELS

Global level	Aspect level
Human-machine environment	DS: decision support system
	HME: human-machine environment
Team	HME: team
	SO: self-organized team
Participant	DS: user
	HME: participant
	SO: participant
	C: entity
User	DS: user
	HME: user
Role	DS: role
	SO: role
Artifact	DS: alternative
	HME: artifact
	SO: solution
Problem	DS: problem
	HME: problem
Task	DS: activity
	SO: task
	C: task
Competency	SO: competency
	C: competency
Context	DS: context
	HME: context

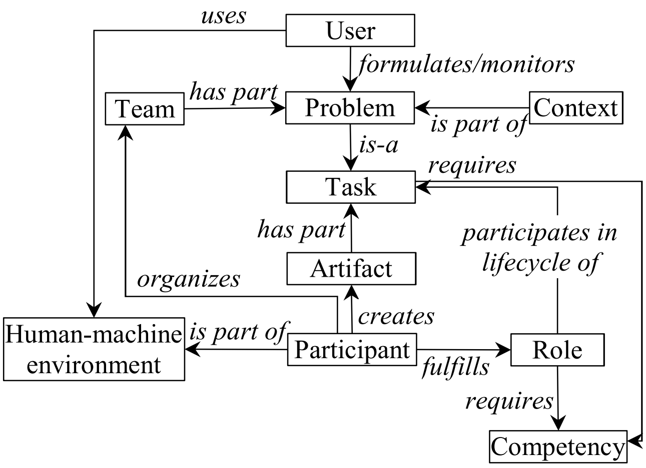

FIGURE 8. **Conceptualization for global level of multi-aspect ontology.**

Besides, the notions of aspects (A) and bridging rules (R) are introduced.

The OWL individuals are supposed to be only local, however, they can be linked with those from other aspects if their classes are linked with corresponding bridging rules.

Hence, the multi-aspect ontology (O) is described as a quadruple:

$$O = (C^G, P^G, A, R).$$

Each aspect (A_i) is described as follows:

$$A_i = \left(C^A_{A_i}, P^A_{A_i}, E^L_{A_i}, R_{A_i} V_{A_i}\right) \in A, i \in [1, |A|],$$

where

$C^A_{A_i}$ – is the set of aspect classes of aspect A_i;

$P^A_{A_i}$ – is the set of aspect properties of aspect A_i;

$E^L_{A_i}$ – is the set of local elements of aspect A_i (representation components of the aspect modeling formalism);

R_{A_i} – is the set of equivalent relationships between elements (classes and properties) of aspect A_i and its local elements: $e^A_j \overset{\equiv}{\rightarrow} e^L_k$, where $e^A_j \in C^A_{A_i} \cup P^A_{A_i}$, $e^L_k \in E^L_{A_i}$;

V_{A_i} – is the set of individuals of aspect A_i.

To illustrate the formalism, in this and the next sections we will consider the following aspects: "*decision support*" (A_{DS}) and "*self-organization*" (A_{SO}).

Three types of bridging rules are introduced based on those from the distributed description logics:

$$G{:}\,c \overset{\sqsubseteq}{\rightarrow} A_i{:}\,d$$

– "into" (or "inclusion in") bridging rule meaning that individuals of a global level (G) element c are individuals of an aspect level element d of aspect A_i, for example, individuals of the classes "*User*" and "*Participant*" of the global level are individuals of the class "*user*" of the aspect "*decision support*": $G{:}\,User \overset{\sqsubseteq}{\rightarrow} A_{DS}{:}\,user$, $G{:}\,Participant \overset{\sqsubseteq}{\rightarrow} A_{DS}{:}\,user$.

$$G{:}\,c \overset{\sqsupseteq}{\rightarrow} A_i{:}\,d$$

– "onto" (or "inclusion of", or "inclusion") bridging rule meaning that an individual of an aspect level element d of aspect A_i is an individual of a global level element c, for example, individuals of the class "*alternative*" of the aspect "*decision support*" are individuals of the class "*Artifact*" of the global level: $G{:}\,Artifact \overset{\sqsupseteq}{\rightarrow} A_{DS}{:}\,alternative$.

$$G{:}\,c \overset{\equiv}{\leftrightarrow} A_i{:}\,d$$

– bidirectional "equivalence" bridging rule, meaning that the sets of individuals of the global level element c and the aspect level element d of the aspect A_i are equal, for example, all individuals of the class "*context*" of the aspect "*decision support*" are individuals of the global level class "*Context*", and all individuals of the global level class "*Context*" are individuals of the class "*context*" of the aspect "*decision support*":

$$G{:}\,Context \overset{\equiv}{\leftrightarrow} A_{DS}{:}\,context.$$

V. APPLICATION OF THE METHODOLOGY IN HMCI-BASED DECISION SUPPORT SYSTEM

A prototype of the HMCI-based DSS implements the functions expected from the HMCI environment. In this section, the prototype is used to demonstrate operating such an environment while resolving a problem from the application domain of "e-tourism".

The developed prototype uses the multi-aspect ontology and implements some ideas of the proposed methodology. In particular, in the shared information space, the knowledge of the team relevant to the problem being solved is organized using the multi-aspect ontology. The knowledge pieces are represented as OWL 2 statements and stored in Blazegraph triple store [63]. To support reactive scenarios, when some software services can react to the changes in the information space, we use SEPA architecture [64], allowing SPARQL-based publish/subscribe mechanisms. Therefore, the software services can subscribe to certain patterns in the problem representation and become active when these patterns emerge

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116870, IEEE Access

in the ontology-based problem description. Human participants interact with this representation via GUI components, hiding many technical details of ontology usage. The structured description of the problem contained in the shared information space is gradually extended and refined by the team. As a result, the progress of a team on the problem resolving is reflected by structural changes of the ontological problem description.

With respect to application domains, the multi-aspect ontology is domain-independent and can represent knowledge of an application domain as an aspect when the corresponding aspect (depending on the particular problem from a given domain) is included. This aspect can be replaced with another one on the "plug-in" basis.

The implementation of the proposed methodology for developing multi-aspect ontologies can be divided into two parts. The first is building the multi-aspect ontology for loosely connected aspects independent on the application domain (these aspects are considered in previous sections of the paper). For this part, the aspect and global (shared) levels do not change from problem to problem, therefore, these levels are built once and the corresponding bridging rules are defined in the multi-aspect ontology used in the prototype. However, each new problem brings a new aspect, specifically, the application domain aspect, to the multi-aspect ontology. The process of including the application domain aspect to the multi-aspect ontology is the second part of the multi-aspect ontology development. The HMCI-based DSS supports this process. The user and/or team members can perform the alignment between the application domain aspect and the aspect level of the multi-aspect ontology.

For illustrative purpose, we consider a problem of finding a museum in a certain region. It should be noted, that this example is an illustration, fulfilling two goals. First, it allows us to explain the process of applying the methodology and the resulting multi-aspect ontology without distracting to the specifics of some problem domain ("e-tourism" is quite familiar to most of the readers). Second, in many "e-tourism" scenarios software services play an important role. Sometimes, a problem can even be fully solved by a service. But, at the same time, partial specification or context refinement are most effectively dealt with by human team members, advocating the use of hybrid human-machine teams.

In the considered scenario the user submits a request in natural language to the HMCI-based DSS that is formulated as follows:

"*Please find a museum in St. Petersburg, Russia dedicated to history*."

A software service that implements natural language processing recognizes the user request and identifies that the system resolves the problem of supporting the user with a recommendation of a museum. Another service looks for a domain ontology that represents information on museums in the internal ontology library. If this library does not contain a suitable ontology, then experts are involved in the search. If the required ontology is not found anywhere, it is developed from scratch. The found and reused ontologies or the developed ontologies are saved in the internal ontology library.

In the present example, the OWL ontology of tourism for a Semantic Web provided by the Protégé ontology library [65] is reused. This ontology among other things defines the *museum* concept and represents concepts and relationships to characterize museums. The ontology becomes the application domain aspect (A_{AD}) referred to as tourism aspect. Its fragment relevant to the example is presented in Fig. 9.

If the domain of the problem formulated in the request is new for the DSS, then experts analyze the tourism aspect with the objective to reveal aspect-level concepts, i.e. they find concepts for which there are semantic mappings to the concepts of other aspect ontologies. In the considered example, two concepts are candidates for the mapping: *museum* and *activity* (cf. Fig. 3, Fig. 5, and Fig. 9). From the viewpoint of the decision support aspect, *museum* is an *alternative*; for the self-organization aspect, *museum* is a *solution* that the self-organized team obtains as the result of task solving. The concept of *activity* represented in the tourism aspect seems to correspond to the concept of *activity* represented in the decision support aspect. However, activity in the decision support aspect and activity in the tourism aspect have different meanings. The activity in the tourism is an activity that a tourist does when visiting places of his/her interests, while in decision support, the activity concerns functions of the decision support system. Thus, only the concept of *museum* is captured as the aspect level concept.

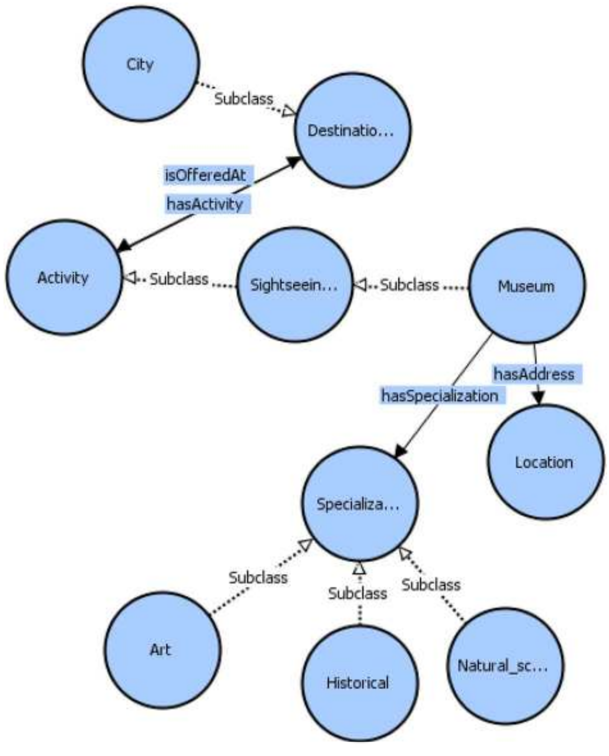

FIGURE 9. Tourism ontology (fragment).

This concept is linked to the concept "*Artifact*" of the global level with the "*inclusion*" bridging rule (Fig. 10):

$G: Artifact \overset{\sqsupseteq}{\rightarrow} A_{AD}: museum.$

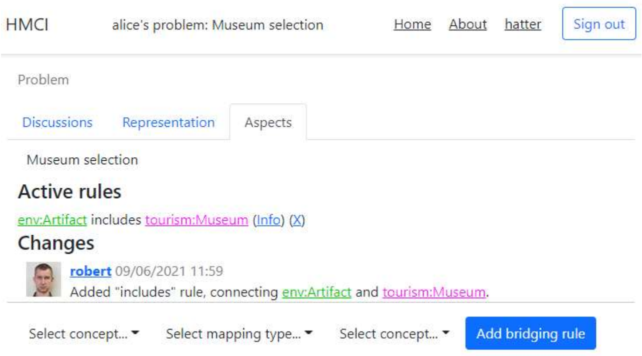

FIGURE 10. Concept mappings.

The alignment relationships between the local-level concepts and aspect level concepts are saved and next time when a problem from the same domain arises, the procedures of the aspect-level aspects identification and the bridging of the aspect and global levels can be performed automatically.

Since the request has some additional information (the location and the area of interest) that was not recognized by the service, and in the tourism ontology, the class "*museum*" has some properties, the HMCI-based DSS tries to automatically map these. In case of failure, the system offer the user to do the matching and suggests to align "*Location*" and "*Specialization*" (the properties of the class "*Museum*") with concepts found in the user request ("*in St. Petersburg, Russia*", "*dedicated to history*"). The user identifies that "*in St. Petersburg, Russia*" is "*has address*" with *location* including "*St. Petersburg, Russia*", and "*dedicated to history*" is "*Specialization*" corresponding to "*Historical*". These patterns are saved in the domain aspect for future use. The clarified information becomes a part of the context.

Below, some of the bridging rules defined for the considered example are presented. One block of the rules represents the structure of the information environment intended to solve the problem formulated in the user request:

a. $G: Human\text{-}machine\ environment \overset{\equiv}{\leftrightarrow} A_{DS}: decision\ support\ system$

b. $G: team \overset{\equiv}{\leftrightarrow} A_{SO}: self\text{-}organized\ team$

The bridging rule "a" means that the HMCI-based DSS solves the problem formulated in the user request as a decision support problem. The rule "b" designates that a self-organized team solves this problem. The global level specifies that participants that are part of the DSS organize the team.

Another block represents terminology for information exchange by the team participants:

c. $G: Artifact \overset{\sqsupseteq}{\rightarrow} A_{DS}: alternative.$

d. $G: Artifact \overset{\sqsupseteq}{\rightarrow} A_{SO}: solution$

e. $G: Artifact \overset{\sqsupseteq}{\rightarrow} A_{AD}: museum.$

The rules "c" and "d" define that the alternatives developed for the considered problem and the problem solution provided by the team are interpreted as artifacts. Particularly, the set of artifacts among other things contains the museum that the DSS recommends to the user (the rule "e").

According to the global level, the DSS's participants resolve the problem formulated in the user request within their roles that require appropriate competencies. Competency matching is used to find participants suitable for a role. For the request in question, participants are expected to have knowledge of museums in St. Petersburg to participate in problem resolving.

First, the required competency is searched for in the participants' profiles. Participants whose profiles include the required competency are invited to be involved in the problem resolving. The aspect ontology of self-organization (Fig. 5) is used when inviting participants. Namely, the HMCI-based DSS provides the participants the information about the purpose of their involvement (to solve the task formulated in the request), roles that the participants are invited to fulfill, and the remuneration. Participants, who are interested in the offer, accept it. If there are several participants interested in the offer, then an efficient team is organized [66].

If no profiles representing the required competency are found, the DSS invites some participant to be the project manager (here project manager is a kind of role). The procedure of the invitation is the same as described above. The project manager becomes responsible for the search and invitation of other participants.

As soon as the team is organized, the participants can start acting to solve the task formulated in the user request. For the considered example, the team has been organized based on the participant profiles. It comprises a software service and a human. A software service sends a query to the tourism ontology, which returns a list of museums that have relationships to the class "*historical*". The list contains more than 50 museums somehow related to different fields of history. The human expert understands that the list has to be narrowed down to a reasonable size. He/she asks the user to specify some additional preferences (e.g., archeology, Russian history, history of St. Petersburg, Russian life, etc.). The user replies with history of St. Petersburg (Fig. 11). The expert recommends Museum of the History of St. Petersburg (Fig. 12).

VI. DISCUSSION

A multi-aspect ontology is a means of interoperability enabling a human-machine environment to operate for decision support. Such an ontology is supposed to integrate the knowledge of multiple loosely-related domains, maintain extendibility with new knowledge as new problems come, and support the organization of teamwork between humans and machines. The ontology development methodologies analyzed [15]–[27] cannot be used directly to create such an ontology. The main reason is that although most of them

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116870, IEEE Access

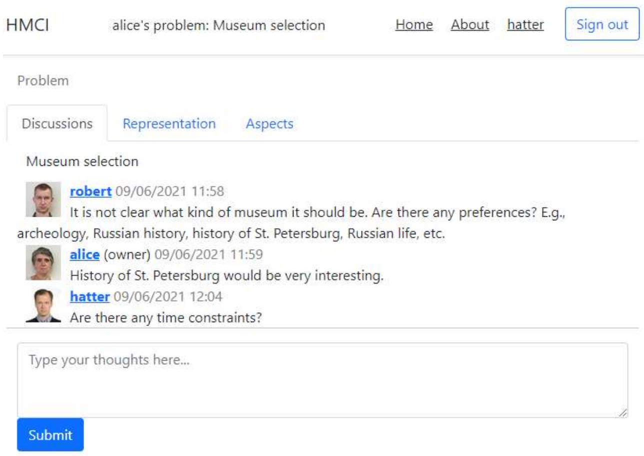

FIGURE 11. Interactions between the user and expert.

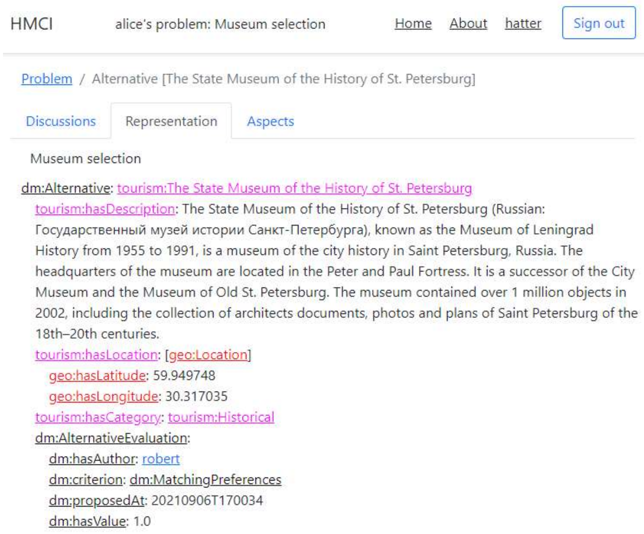

FIGURE 12. Recommending decision (ontology-based representation).

suppose knowledge integration, none offers techniques to implement this. The proposed methodology for the multi-aspect ontology development fills this gap by incorporating an approach to integration of multiple aspects.

In comparison with related approaches that focus on the multi-aspect problem representation (Table V), the aspect integration approach encompasses the best features of them. Regarding the multi-aspect ontology requirements the approach:

– supports the integration of loosely-connected aspects, as opposed to, for example, the multi-viewpoint ontology approach [42], [43] focused on integrating the closely related viewpoints on a problem domain;

– allows integrating knowledge represented by different formalisms unlike the multilingual [32]–[34] or granular ontologies [40], [41] at that preserving formalisms that are beneficial for knowledge representation in aspects;

– allows new knowledge to be introduced without multiple re-alignments in contrast to the model-driven framework [30].

The proposed methodology was used for developing a multi-aspect ontology enabling humans and machines to organize their teamwork for decision support based on human-machine collective intelligence. In the prototype of an HMCI-based decision support system intended for decision support in the "e-tourism" domain, this ontology serves well for ensuring interoperability between humans and machines, and their organization in teams. The latter is achieved by incorporating aspects so that their integrated knowledge constitutes a representation of the problem of human-machine teams self-organization with the decision support purpose.

In addition to the above, the multi-aspect ontology reduces the number of ontology modifications caused by changes in its aspects: it is enough to check the existing relationships (alignments and bridging rules) for the changed aspect from the aspect level up to the global level and introduce corresponding revisions. The high level of independence of aspects makes them autonomous and allows one to introduce changes into one aspect without affecting the others. For instance, when the user introduces the museum selection problem to the prototype, only 1 class (*museum*) from the application domain aspect is introduced into the ontology to indicate that museums are alternatives. Without the multi-aspect ontology this would require integration of the entire Tourism ontology (61 classes and 25 object properties) and evaluate the entire resulting system ontology (about 110 classes and 85 relationships) for its consistency.

TABLE V
COMPARISON OF APPROACHES TO MULTI-ASPECT PROBLEM REPRESENTATION

Approach	Support of heterogeneous formalisms	Adaptable shared ontology level	Maintenance of changes in aspects	Automatic integration	Support of loosely connected aspects	Openness for new knowledge
Model-driven interoperability framework [30]	+	+/-	-	-	+	+/-
Multilingual ontologies [32]–[34]	-	+/- (depending on the approach)	+	+/- (depending on the approach)	-	+
Granular ontologies [40], [41]	-	+/-	+/-	+/- (depending on the approach)	+/-	+/-
Multi-viewpoint ontology [42], [43]	+	+	+	-	-	+
Multi-aspect ontologies	+	+	+	+/-	+	+

The main limitation of the proposed methodology is the manual ontology alignment. Such an alignment requires significant efforts and existing automatic techniques produce high-quality results only within narrow domains. At the same time, for the knowledge of application domains the manual alignment is the only way so far, since kinds of alignments between the application domain aspects and the multi-aspect ontology levels depend on the decision support problem, and each problem requires specific alignments. Nevertheless, the methodology supports automatic alignment of the local and aspect levels for recurring domains with subsequent automatic bridging the aspect and global levels. This approach seems efficient since it reduces the need for mappings (both alignment and bridging) and at the same time takes into account specifics of different decision support problems.

The methodology for the development of multi-aspect ontologies can be used to develop ontologies of complex knowledge-based systems that operate with knowledge from multiple loosely-connected domains. The developed multi-aspect ontology of a human-machine environment intended to support decisions is beneficial to achievement of teamwork between humans and machines while supporting decisions.

VII. CONCLUSION

A four-stage methodology for the development of multi-aspect ontologies is proposed. It adopts an ontology development pattern followed by most ontology development methodologies and extends it with an aspect integration approach. The methodology meets the requirement of the multi-level ontology structure and provides principles for multi-level ontology integration, which is often found in complex systems dealing with multiple domains.

The proposed methodology has been applied to the development of a multi-aspect ontology for decision support based on human-machine collective intelligence. The developed ontology integrates aspects of decision support, human-machine environment, self-organization, competency, and application domain.

The prototype of an HMCI-based decision support system intended for decision support in the “e-tourism” domain proves the validity of the multi-aspect ontology developed based on the proposed methodology.

Future research is aiming to experiments with the prototype to support decisions on complex problems that require knowledge from several application domains, which includes corresponding updates of the multi-aspect ontology.

ACKNOWLEDGEMENT

The research is funded by the Russian Science Foundation (project # 19-11-00126), except for the competence aspect developed within the State Research, project number 0073-2019-0005.

REFERENCES

[1] J. C. Glenn, “Collective Intelligence and an Application by The Millennium Project,” *World Futures Review*, vol. 5, no. 3, pp. 235–243, Sep. 2013, doi: 10.1177/1946756713497331.
[2] Y. Zhang, “CityMatrix – An Urban Decision Support System Augmented by Artificial Intelligence,” Massachusetts Institute of Technology, 2017.
[3] G. Willcox, L. Rosenberg, D. Askay, L. Metcalf, E. Harris, and C. Domnauer, “Artificial Swarming Shown to Amplify Accuracy of Group Decisions in Subjective Judgment Tasks,” in *Advances in Information and Communication*, vol. 70, K. Arai and R. Bhatia, Eds. Cham: Springer, 2020, pp. 373–383.
[4] A. Berditchevskaia and P. Baeck, “The future of minds and machines,” London, 2020. [Online]. Available: https://media.nesta.org.uk/documents/FINAL_The_future_of_minds_and_machines.pdf.
[5] S. G. Pease *et al.*, “An interoperable semantic service toolset with domain ontology for automated decision support in the end-of-life domain,” *Future Generation Computer Systems*, vol. 112, pp. 848–858, Nov. 2020, doi: 10.1016/j.future.2020.06.008.
[6] D. Roosan, A. Hwang, A. V Law, J. Chok, and M. R. Roosan, “The inclusion of health data standards in the implementation of pharmacogenomics systems: a scoping review,” *Pharmacogenomics*, vol. 21, no. 16, pp. 1191–1202, Nov. 2020, doi: 10.2217/pgs-2020-0066.
[7] G. Zacharewicz, N. Daclin, G. Doumeingts, and H. Haidar, “Model Driven Interoperability for System Engineering,” *Modelling*, vol. 1, no. 2, pp. 94–121, Oct. 2020, doi: 10.3390/modelling1020007.
[8] A. Smirnov and A. Ponomarev, “Decision Support Based on Human-Machine Collective Intelligence: Major Challenges,” in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, vol. 11660, Heidelberg: Springer, 2019, pp. 113–124.
[9] A. Smirnov and A. Ponomarev, “Human-Machine Collective Intelligence Environment for Decision Support: Conceptual and Technological Design,” in *2020 27th Conference of Open Innovations Association (FRUCT)*, Sep. 2020, pp. 253–259, doi: 10.23919/FRUCT49677.2020.9211077.
[10] A. Smirnov, T. Levashova, N. Shilov, and A. Ponomarev, “Multi-aspect Ontology for Interoperability in Human-machine Collective Intelligence Systems for Decision Support,” in *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 2019, pp. 458–465, doi: 10.5220/0008356304580465.
[11] K. Sandkuhl, N. Shilov, and A. Smirnov, “Facilitating digital transformation: success factors and multi-aspect ontologies,” *International Journal of Integrated Supply Management*, vol. 13, no. 4, p. 376, 2020, doi: 10.1504/IJISM.2020.110739.
[12] N. Shilov and N. Teslya, *Ontology-Based Fragmented Company Knowledge Integration: Multi-aspect Ontology Building*, vol. 373 LNBIP. 2019.
[13] A. Smirnov, N. Shilov, and V. Parfenov, “Building a Multi-aspect Ontology for Semantic Interoperability in PLM,” *IFIP Advances in Information and Communication Technology*, vol. 565, pp. 107–115, 2019, doi: 10.1007/978-3-030-42250-9_10.
[14] A. Smirnov, T. Levashova, A. Ponomarev, and N. Shilov, “Methodology for Multi-aspect Ontology Development,” *Lecture Notes in Business Information Processing*, vol. 414, pp. 97–109, 2021, doi: 10.1007/978-3-030-73976-8_8.
[15] M. Uschold and M. King, “Towards a methodology for building ontologies,” 1995, doi: 10.1.1.697.8733.
[16] M. F. Lopez, A. Gomez-Perez, J. P. Sierra, and A. P. Sierra, “Building a chemical ontology using Methontology and the Ontology Design Environment,” *IEEE Intelligent Systems*, vol. 14, no. 1, pp. 37–46, Jan. 1999, doi: 10.1109/5254.747904.
[17] M. Fernandez, A. Gomes-Perez, and N. Juristo, “METHONTOLOGY: From Ontological Art Towards Ontological Engineering,” in *AAAI Proceedings of the Symposium on Ontological Engineering*, 1997, pp. 33–40.
[18] N. F. Noy and D. L. McGuinnnes, “Ontology Development 101: A Guide to Creating Your First Ontology,” 2001, [Online]. Available: http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html.
[19] N. F. Noy and S. Tu, “Ontology Engineering for the Semantic Web and Beyond.” American Medical Informatics Association, 2003, [Online]. Available: https://slideplayer.com/slide/12403284/.
[20] S. Staab, R. Studer, H.-P. Schnurr, and Y. Sure, “Knowledge processes and ontologies,” *IEEE Intelligent Systems*, vol. 16, no. 1, pp. 26–34, Jan. 2001, doi: 10.1109/5254.912382.
[21] Y. Sure, S. Staab, and R. Studer, “On-To-Knowledge Methodology (OTKM),” in *Handbook on Ontologies*, Berlin, Heidelberg: Springer

Berlin Heidelberg, 2004, pp. 117–132.
[22] F. Neuhaus *et al.*, "Towards ontology evaluation across the life cycle," *Applied Ontology*, vol. 8, no. 3, pp. 179–194, 2013, doi: 10.3233/AO-130125.
[23] M. C. Suárez-Figueroa, A. Gómez-Pérez, and M. Fernández-López, "The NeOn Methodology framework: A scenario-based methodology for ontology development," *Applied Ontology*, vol. 10, no. 2, pp. 107–145, Sep. 2015, doi: 10.3233/AO-150145.
[24] A. Abdelghany, N. Darwish, and H. Hefni, "An Agile Methodology for Ontology Development," *International Journal of Intelligent Engineering and Systems*, vol. 12, no. 2, pp. 170–181, Apr. 2019, doi: 10.22266/ijies2019.0430.17.
[25] A. Sattar, M. N. Ahmad, E. S. M. Surin, and A. K. Mahmood, "An Improved Methodology for Collaborative Construction of Reusable, Localized, and Shareable Ontology," *IEEE Access*, vol. 9, pp. 17463–17484, 2021, doi: 10.1109/ACCESS.2021.3054412.
[26] G. Guizzardi, G. Wagner, J. P. A. Almeida, and R. S. S. Guizzardi, "Towards ontological foundations for conceptual modeling: The unified foundational ontology (UFO) story," *Applied Ontology*, vol. 10, no. 3–4, pp. 259–271, Dec. 2015, doi: 10.3233/AO-150157.
[27] M. Gruninger and M. S. Fox, "Methodology for the Design and Evaluation of Ontologies," 1995, [Online]. Available: http://www.eil.utoronto.ca/wp-content/uploads/enterprise-modelling/papers/gruninger-ijcai95.pdf.
[28] A. Asmae, S. Souhail, Z. El Moukhtar, and B. Hussein, "Using ontologies for the integration of information systems dedicated to product (CFAO, PLM…) and those of systems monitoring (ERP, MES..)," in *2017 International Colloquium on Logistics and Supply Chain Management (LOGISTIQUA)*, Apr. 2017, pp. 59–64, doi: 10.1109/LOGISTIQUA.2017.7962874.
[29] C. Palmer, E. N. Urwin, R. I. M. Young, and E. Marilungo, "A reference ontology approach to support global product-service production," *International Journal of Product Lifecycle Management*, vol. 10, no. 1, p. 86, 2017, doi: 10.1504/IJPLM.2017.083003.
[30] M. Lafleur, W. Terkaj, F. Belkadi, M. Urgo, A. Bernard, and M. Colledani, "An Onto-Based Interoperability Framework for the Connection of PLM and Production Capability Tools," in *PLM 2016: Product Lifecycle Management for Digital Transformation of Industries*, 2016, pp. 134–145, doi: 10.1007/978-3-319-54660-5_13.
[31] J. Euzenat, J. David, and M. Atencia, "EDOAL: Expressive and Declarative Ontology Alignment Language." https://moex.gitlabpages.inria.fr/alignapi/edoal.html.
[32] M. Espinoza, A. Gómez-Pérez, and E. Montiel-Ponsoda, "Multilingual and Localization Support for Ontologies," in *The Semantic Web: Research and Applications. ESWC 2009*, vol. 5554, L. Aroyo, P. Traverso, F. Ciravegna, P. Cimiano, E. Hyvönen, R. Mizoguchi, E. Oren, M. Sabou, and E. Simperl, Eds. Berlin, Heidelberg: Springer, 2009, pp. 821–825.
[33] M. Florrence, "Building a multilingual ontology for education domain using monto method," *Computer Science and Information Technologies*, vol. 1, no. 2, pp. 47–53, Jul. 2020, doi: 10.11591/csit.v1i2.p47-53.
[34] T. Ivanova, "A bilingual ontology mapping and enrichment approach for domain ontologies in e-learning," in *Proceedings of the 20th International Conference on Computer Systems and Technologies*, Jun. 2019, pp. 284–291, doi: 10.1145/3345252.3345257.
[35] E. Montiel-Ponsoda, G. Cea, A. Gomez-Perez, and W. Peters, "Modelling Multilinguality in Ontologies," in *COLING 2008, 22nd International Conference on Computational Linguistics, Posters Proceedings*, 2008, pp. 67–70, [Online]. Available: https://www.aclweb.org/anthology/C08-2017/.
[36] J. Salim, S. F. Mohamad Hashim, and S. A. Mohamad Noah, "Symbiosis of thesaurus, domain expert and reference sources in designing a framework for the construction of a multilingual ontology for Islamic Portal," *International Journal on Electrical Engineering and Informatics*, vol. 4, no. 1, pp. 120–133, Mar. 2012, doi: 10.15676/ijeei.2012.4.1.10.
[37] B. Lauser, T. Wildemann, A. Poulos, F. Fisseha, J. Keizer, and S. Katz, "A Comprehensive Framework for Building Multilingual Domain Ontologies: Creating a Prototype Biosecurity Ontology," in *Proceedings of International Conference on Dublin Core and Metadata for e-Communities*, 2002, pp. 113–123.
[38] E. Nichols, F. Bond, T. Tanaka, S. Fujita, and D. Flickinger, "Multilingual Ontology Acquisition from Multiple MRDs," in *Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, 2006, pp. 10–17, [Online]. Available: https://www.aclweb.org/anthology/W06-0502.
[39] A. Al Azemi, S. Nefti, U. Manzoor, and Y. Rezgui, "Building a Bilingual Bio-Ontology Platform for Knowledge Discovery," *International Journal of InnovativeComputing, Information and Control*, vol. 7, no. 12, pp. 7067–7075, 2011.
[40] S. Calegari and D. Ciucci, "Granular computing applied to ontologies," *International Journal of Approximate Reasoning*, vol. 51, no. 4, pp. 391–409, Mar. 2010, doi: 10.1016/j.ijar.2009.11.006.
[41] M. Priya and C. Aswani Kumar, "An approach to merge domain ontologies using granular computing," *Granular Computing*, vol. 6, no. 1, pp. 69–94, Jan. 2021, doi: 10.1007/s41066-019-00193-3.
[42] M. Hemam and Z. Boufaïda, "MVP-OWL: a multi-viewpoints ontology language for the Semantic Web," *International Journal of Reasoning-based Intelligent Systems*, vol. 3, no. 3/4, p. 147, 2011, doi: 10.1504/IJRIS.2011.043539.
[43] O. Djama and Z. Boufaida, "Instantiation of the multi-viewpoints ontology from a resource," *International Journal of Computers and Applications*, pp. 1–12, Jan. 2020, doi: 10.1080/1206212X.2020.1711615.
[44] K. L. Guo, "DECIDE: a decision-making model for more effective decision making by health care managers," *The Health Care Manager*, vol. 27, no. 2, pp. 118–127, 2008, doi: 10.1097/01.HCM.0000285046.27290.90.
[45] P. Brown, "Career coach: decision-making," *Pulse*, no. 29 November, 2007, [Online]. Available: http://www.pulsetoday.co.uk/career-coach-decision-making/10967084.article.
[46] H. Simon, *The New Science of Decision-Making*. New York: Harper and Row, 1960.
[47] H. Simon, "Rational Decision Making in Business Organizations," *American Economic Association*, vol. 69, no. 4, pp. 493–513, 1979, [Online]. Available: http://www.jstor.org/stable/1808698%0D.
[48] C. Heath and D. Heath, "Decisive: the WRAP process," 2013. https://heathbrothers.com/ot/wp-content/uploads/2013/03/The_WRAP_Process_one_pager.pdf (accessed Jan. 17, 2020).
[49] C. Argyris, *Management and organizational development: The path from XA to YB*. New York: McGraw-Hill, 1971.
[50] J. Rest, D. Narvaez, M. Bebeau, and S. Thoma, "A Neo-Kohlbergian approach: The DIT and schema theory," *Educational Psychology Review*, vol. 11, no. 4, pp. 291–324, 2009, doi: 10.1023/A:1022053215271.
[51] "7 steps to effective decision making," *UMass Dartmouth*. https://www.umassd.edu/media/umassdartmouth/fycm/decision_making_process.pdf (accessed Jan. 29, 2020).
[52] H. Hörmann, "FOR-DEC: A prescriptive model for aeronautical decision making," in *Human Factors in Aviation Operations*, R. Fuller, J. Neil, and N. McDonald, Eds. Farnham: Avebury Aviation, 1995, pp. 17–23.
[53] A. J. Walters, *Crew resource management is no accident*. Wallingford: Aries, 2002.
[54] G. A. Klein, *Sources of Power: How People Make Decisions*. Cambridge: MIT Press, 1998.
[55] V. H. Vroom and A. A. Jago, *The New Leadership: Managing Participation in Organizations*. Englewood Cliffs: Prentice-Hall, 1988.
[56] J. Rockwell, I. R. Grosse, S. Krishnamurty, and J. C. Wileden, "A Decision Support Ontology for collaborative decision making in engineering design," in *2009 International Symposium on Collaborative Technologies and Systems*, 2009, pp. 1–9, doi: 10.1109/CTS.2009.5067456.
[57] J. A. Rockwell, I. R. Grosse, S. Krishnamurty, and J. C. Wileden, "A Semantic Information Model for Capturing and Communicating Design Decisions," *Journal of Computing and Information Science in Engineering*, vol. 10, no. 3, Sep. 2010, doi: 10.1115/1.3462926.
[58] P. Nowara, "Sample Decision Ontology," *W3C Incubator Group*

Report, 2012. https://www.w3.org/2005/Incubator/decision/XGR-decision-20120417/Sample_Decision_Ontology.html (accessed Apr. 14, 2020).

[59] A. K. Dey, "Understanding and Using Context," *Personal and Ubiquitous Computing*, vol. 5, no. 1, pp. 4–7, Feb. 2001, doi: 10.1007/s007790170019.

[60] D. L. McGuinness and F. van Harmelen, "OWL Web Ontology Language Overview," *W3C Recommendation*, 2004. https://www.w3.org/TR/owl-features/ (accessed Mar. 25, 2020).

[61] "Protégé." https://protege.stanford.edu/.

[62] M. Hemam, "An Extension of the Ontology Web Language with Multi-Viewpoints and Probabilistic Reasoning," *International Journal of Advanced Intelligence Paradigms*, vol. 10, no. 1, p. 1, 2018, doi: 10.1504/IJAIP.2018.10003857.

[63] "Blazegraph: The Official Webpage." https://blazegraph.com/.

[64] L. Roffia *et al.*, "A Semantic Publish-Subscribe Architecture for the Internet of Things," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1274–1296, Dec. 2016, doi: 10.1109/JIOT.2016.2587380.

[65] "Protege: Welcome to the Protege Ontology Library!" https://protegewiki.stanford.edu/wiki/Protege_Ontology_Library (accessed Mar. 30, 2020).

[66] A. Smirnov and A. Ponomarev, "Stimulating Self-Organization in Human-Machine Collective Intelligence Environment," *2021 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)*. IEEE, pp. 94–102, 2021, doi: 10.1109/CogSIMA51574.2021.9475937.