An Overview of Ontology Engineering Methodologies in the Context of Public Administration

Bernd Stadlhofer, Peter Salhofer, Augustin Durlacher
Institute of Information Management
FH JOANNEUM, University of Applied Sciences
Graz, Austria
bernd.stadlhofer@fh-joanneum.at, peter.salhofer@fh-joanneum.at, augustin.durlacher@fh-joanneum.at

Abstract— **This paper gives an overview of latest ontology engineering methodologies that are analyzed in terms of a representative set of criteria and aspects. The portfolio of criteria considers general structural aspects of ontology development (such as strategy for building ontologies) as well as project management aspects (such as recommended process model or the consideration of collaborative construction). While the study criteria principally stay generic we particularly try to include possible characteristics of the E-Government domain. Whereas the study shows that none of the discussed methodologies is fully mature to serve as a domain expert centered ontology engineering methodology in the context of electronic service provisioning in public administration, it also outlines the potential of the discussed methodologies to which extend they can contribute to a new methodology in this field.**

Keywords-Ontology Engineering Methodologies; E-Government; Comparative Study.

I. Introduction

In the context of public administration, ontology engineering is mainly involved when applying Semantic Web technologies to E-Government and to the electronic public service provisioning process respectively. Actually, this domain has become an important field of research aiming at enhancing transparency, interoperability as well as citizen orientation of public agencies [1]. In fact, developing formal ontologies is a complex task that requires having significant skills in software and knowledge engineering in order to being able to design, implement, and maintain ontologies. Beyond this, it requires domain expertise in order to verify the correctness of domain specific ontologies. Whereas a domain expert (in the context of public administration we use the term domain expert synonymously with legal expert) possesses in-depth knowledge of the specific domain to be modeled she or he is very likely not to have sufficient ontology engineering skills at the same time. In order to reduce the complexity of ontology engineering for domain experts methodological guidelines assisted by intelligent tooling have to be applied.

When reviewing literature on semantic E-Government initiatives listed in [1], only one initiative (i.e., [2]) mentions explicitly a specific ontology engineering methodology for designing semantic models. Hence, relevant ontologies in the E-Government sector still tend to be built rather on an ad hoc basis than following a well-defined engineering process supported by adequate tools (similar observations are documented in [3]). As a consequence, the actual ontology engineering is rather done by software engineers than by domain experts. This circumstance also fosters the effect that E-Government projects, generally, suffer from unsustainable activities in the organizational environment, e.g., external stakeholders (such as knowledge and software engineers) leave after a project ends [4].

An essential aspect in the context of public administration is the consideration of legal certainty. In fact, in ontology engineering for the public administration sector, legislation and enforcement of law on all governmental levels has to be ensured. This requires collaboration with a variety of different legal experts. In many cases, constraints probably might have to be weakly encoded, supported by textual explanations and links to further information and supporting bodies. This is necessary to reflect the special demands of a legal system and to safeguard legal certainty [5]. An ontology engineering methodology in the context of public administration should explicitly include steps for domain experts that deal with this circumstance. Hence, not only the validation of the formal model consistency has to be considered but also steps for a simple validation of legal aspects have to be applied.

Consequently, it is our claim to support domain experts of public administration with sufficient guidelines, which enables them to design, implement, verify and maintain their semantic artifacts by themselves. A first step towards this goal is to review the state of the art in this field of research. Hence, the aim of this paper is to give an overview of general ontology engineering methodologies available, having a focus on impacts on E-Government in particular. The resulting overview should give valuable input to our overall research goal, which is to establish a "domain expert centered ontology engineering methodology in the context of electronic service provisioning in public administration".

The remainder of this paper is organized as follows: In Section II, the specific criteria and aspects for the study are introduced followed by a description of a selected list of methodologies in terms of the presented analysis aspects in Section III. Section IV briefly compares the selected methodologies according to the chosen criteria. Section V lists related work in the field of evaluating ontology engineering methodologies. In Section VI, the work is concluded with the summarization of the analysis' results.

II. ANALYSIS ASPECTS

In this section, the concrete aspects and criteria considered for the study are discussed. The portfolio of criteria considers general structural aspects of ontology development (such as strategy for building ontologies) as well as project management aspects (such as recommended life cycle) and aspects directly related to the E-Government domain (such as collaborative construction among public authorities or consideration of legal certainty):

1) Strategy for building ontologies: With this aspect it is examined which strategy is used to develop ontologies. Is it, a) application-dependent, which means that resulting ontologies are designed for and usable by a specific semantic application only, b) application-independent, which means reuse of resulting ontologies is maximized by developing general-purpose descriptions, or c) application-semidependent, where possible scenarios of ontology use are somehow limited [6]. Generally, concerning the strategy for building ontologies it can be argued that "… the more an ontology is independent of application perspectives, the less usable it will be. In contrast, the closer an ontology is to application perspectives, the less reusable it will be" [8]. Thus, there is always a trade-off between application-dependent approaches that typically add some extra value to ontologies, since they can be immediately used in a particular context, and more general application-independent strategies that allow for simplified reuse in different contexts.

2) Recommended process: This aspect examines the existence or recommendation of specific process models one has to go through in order to model ontologies, e.g., being aligned along the general waterfall phases or following iterative, cyclical or agile development models.

3) Consideration of collaborative construction: Modeling an ontology of some public administration domain generally requires numerous authorities to be involved. Since these experts from different public agencies are typically locally distributed, it is simply not possible to develop and maintain all relevant information at one central point [5], particularly when the modeled domain is rather complex.

4) Tool-support: Does a specific tool explicitly support the methodology in question? In terms of enabling domain experts to model ontologies, providing context is a major goal in order to reduce complexity of the modeling process [9]. Specific methodological guidelines combined with intelligent and human-centered tooling should overcome a possible lack in engineering skills.

5) Target group: For what group of people is the methodology primarily designed? Traditionally an ontology engineering methodology is intended for knowledge and ontology engineering experts whereas domain experts are only involved in the knowledge elicitation phase. In contrast, we consider domain experts as the primary target group that should have a maximum of responsibility during the whole ontology life cycle. Only domain experts possess the respective knowledge to be modeled and the expertise to ensure legal certainty of resulting artifacts. Centralizing domain experts in the ontology engineering process should also boost sustainable development of semantic initiatives in E-Government.

Whereas aspects 1 and 2 rather aim at enabling a structured discussion of the selected methodologies, aspects 3 to 5 represent methodological requirements, which have been identified by conducting a number of expert interviews with representatives of public agencies on municipal and federal level in Austria. The first three aspects have been derived from existing comparative studies (i.e., [6]). However, what is new in our approach is to specifically focus on analyzing aspects of human-centered computing in ontology engineering. Beyond these 5 aspects, we initially identified some more (e.g., reuse), however, analysis showed that the investigated methodologies could not be differentiated along these aspects. According to Lutz and Stelzer [7], only criteria that enable differentiations between the target objects should be used in comparative studies. This is why such aspects have been removed from the final analysis.

III. METHODOLOGIES

Literature research resulted in a list of 20 documented ontology engineering methodologies mainly reported by Casellas [10]. The methodology developed by Uschold and King [11] can be considered the first approach towards developing a methodology for building ontologies. This methodology builds the foundation for many other approaches that have emerged over the last couple of years. In this study we discuss a selected list of these methodologies based on meeting the requirements represented by aspects 3 to 5 (c.f. Section II).

Initially, we planned to exclude all methods that do not meet all of these 3 requirements to the following extend.

- Aspect 3: There have to be at least recommendations for a collaborative development process.
- Aspect 4: There is at least one modeling tool available that explicitly supports the methodology.
- Aspect 5: The methodology has to focus on the domain expert as the major target group in the modeling process.

In fact, one single methodology in question (i.e., Methodology 4) fulfills all three of these requirements in a reasonable way. Consequently, we revised the exclusion criteria that to be included in the study the methodology under question at least has to address one out of these three methodological requirements.

A. Methodology 1

Holsapple and Joshi [12] present a "Collaborative Approach to Ontology Design". The authors discuss the fundamental importance of ontological commitment, which is "…the agreement by multiple parties (persons and software systems) to adopt a particular ontology when communicating about the domain of interest. … Working toward ontological commitment should not be an afterthought, but rather an integral aspect of ontological engineering. This contention underlies the collaborative approach to ontology design we advocate." [12].

The methodology suggests four phases in the ontology engineering process (aspect 2): Preparation, Anchoring, Iterative Improvement, and Application. In the preparation phase design criteria are defined, boundary conditions and evaluation standards are determined. This aims at both, guiding development of the ontology and assessing the degree of its success. In the anchoring phase an initial version of the ontology is created serving as an anchor to help focus the attention of collaborators. In phase 3 the approach uses an adaption of the Delphi method, which is a formal technique for integrating the individual opinions of a group of experts on some topic. "This gives a systematic way for gathering perspectives and critiques on an ontology as a basis for iterative improvement" [12]. Finally, in phase 4, the ontology is explored in various ways in order to prove the ontology utility. Thereby, the authors do not report about a concrete dependence on a specific semantic application, which leads to an application-independent methodology (aspect 1).

By using the Delphi method a clear and structured collaboration process is introduced. However, this form of collaboration tends to be rather inflexible and heavyweight as feedback collection is coordinated centrally by a control board and not interactively and immediately shared by all participants. The validation of legal certainty is not explicitly addressed. This can be defined as evaluation standard in phase 1 and therefore also be included in iterative Delphi rounds, though (aspect 3). As the name suggests the methodology concentrates on the collaboration aspect only and does not include any guidelines concerning the actual modeling process, which is therefore naturally conducted by classical ontology engineers (aspect 5). Tool-support does not exist (aspect 4).

B. *Methodology 2*

DIstributed, Loosely-controlled and evolvInG Engineering of oNTologies (DILIGENT) represents a methodology that focuses on the evolution of ontologies instead of the initial design. Thus, the methodology supports an evolutionary lifecycle (aspect 2). It focusses on user-centric ontology development and provides integration of automatic agents in the process of ontology evolution [13].

The process starts with various stakeholder-groups (domain experts, users, knowledge engineers, ontology engineers) building together an initial version of the ontology. The initial version results from a rather quick consensus about some high-level terms among all participants. Subsequently, users start to work with the ontology and locally adapt (by sub-classing) it to their specific needs. A control board collects change requests to the shared core ontology. The control board then analyses the various local ontologies, tries to find similarities and introduces a new version of the shared ontology. The control board also regularly revises the shared ontology in terms of not diverging too far. Ontology engineers are responsible for maintaining the ontology based on the board's decisions. Users can then locally update the local ontologies in terms of reusing new terms instead of using their previously defined local terms [13].

With this approach reuse should be maximized among all users whereas not narrowing usage in different application scenarios (aspect 1).

The authors argue that decentralized knowledge management systems are getting increasingly important and therefore emphasize distributed and collaborative construction (aspect 3).

Domain experts in a distributed setting are supported by a fine-grained methodological approach based on the Rhetorical Structure Theory [14]. A standard Wiki is used to allow a traceable discussion. Snapshots of the ontology agreed on are imported to the Wiki, in order to visualize the ontology and ease the discussion of it (aspect 4).

DILIGENT involves numerous different user groups in the engineering process, namely domain experts, users, knowledge engineers and ontology engineers. In the revision phase domain experts are responsible for evaluating an ontology from a domain point of view (does it represent the domain, or does it contain factual errors?). This may also include the validation of legal certainty as necessary for public administration. In fact, the methodology was also applied at the development of an ontology for professional legal knowledge [15]. However, the actual ontology implementation is still intended for ontology engineers (aspect 5).

C. *Methodology 3*

Very similar to DILIGENT (Section III.B), the Human-Centered Ontology Engineering Methodology (HCOME) [16] supports the development and evaluation of "living" ontologies in the context of communities of knowledge workers. The authors mention common impediments for knowledge workers (or domain experts) to participate actively in ontology engineering: they are unfamiliar with formal representation languages and knowledge engineering principles as well as with methods and techniques for constructing and synthesizing ontologies. The main goal of HCOME therefore is to empower domain experts to evolve their formal conceptualizations in their day-to-day activities. Thus, this methodology focuses on the active participation of domain experts in the ontology life cycle (aspect 5). For this purpose, the authors also developed a Human Centered Ontology Engineering Environment (HCONE), which directly supports the development of ontologies following the HCOME methodology (aspect 4).

The methodology proposes specification, conceptualization and exploitation as the three life cycle phases of ontology engineering. All involved tasks are performed iteratively, until a consensus has been reached between the participants (aspect 2). In the specification phase, knowledge workers are joining groups aimed at developing shared ontologies. Workers are discussing requirements in a shared space, produce documents and agree on the aim and the scope of a new ontology [16]. Consequently, the aspect of collaborative development is directly addressed by this methodology (aspect 3). In the conceptualization phase, workers can follow any approach to the development of ontologies in their personal space. In the exploitation phase shared ontologies can be used in the context of specific ontology-driven applications and settings. However, the overall methodology is application-independent, as it doesn't give a recommendation for a specific semantic application to use (aspect 1). The evaluation and further development of personal ontologies are achieved via a recorded structured conversation in order to enable the tracking of changes and decisions.

D. Methodology 4

The "Integrated Modeling Methodology" [17] principally guides the process of creating application domain dependent parts of an organizational learning system named Advanced Process- Oriented Self- Directed Learning Environment (APOSDLE). The methodology consists of four main phases: Scope & Boundaries, Knowledge Acquisition, Modeling of Domain, and Modeling of Learning Goals. Validation & Revision is included as individual activity in all of the main phases (aspect 2). The resulting semantic artifacts are directly applied and exploited in the APOSDLE system [26], which leads to an application-dependent approach (aspect 1). Domain experts are considered to be an important stakeholder group and mostly included in the knowledge acquisition phase. The knowledge acquisition is performed with well-known state-of-the-art techniques like, interviews, card sorting, laddering, and concept/step/section listing. The authors thereby mention the problem that domain experts are often rarely available and scarcely motivated towards modeling [17] (aspect 5).

The methodology is explicitly supported by the so-called Modeling WiKi (MoKi), which allows users to describe semantic artifacts in an informal but structured manner using natural language. The subsequent automatic translation into formal models does not require the users to have in-depth formal modeling skills (aspect 4). The Wiki nature of the MoKi naturally enables a collaborative tool that provides support for domain experts with hardly any knowledge engineering skills to model domains directly. However, the methodology suggests that domain experts, knowledge engineers and experts (coaches) collaboratively work in a rather agile modeling process (aspect 3).

E. Methodology 5

Klischewski and Ukena [2] present a methodology that aims at the design of semantic E-Government services driven by user requirements. The authors suggest a step-by-step design process that signals public administration authorities to focus on the intended common understanding of citizens concerning the description of public administration services' interfaces. Generally, the authors describe the aim of the design of semantic structures in E-Government as: to support informational needs during service processing, to capture domain knowledge and to support technical implementation.

In contrast to other approaches that focus on knowledge-driven or domain-driven design, this methodology focuses on requirements-driven design that should emphasize what users or providers will consider as valuable information [2].

The proposed seven steps for the development of semantic E-Government services are: Identify informational needs, identify required information quality, create glossary of topics and terms, create controlled vocabulary, group and relate terms, design an ontology, implement semantics (aspect 2). While these steps themselves are generic, the authors also give some concrete examples how the specifics of E-Government are addressed.

As already mentioned, the methodology itself is rather generic. Nevertheless the authors use Web Service Modeling Ontology (WSMO) [27] as semantic execution environment in their pilot scenario. However, the authors do not exclude any other semantic execution environments as, e.g., Web Ontology Language for Web Services (OWL-S) [28], which leads to an application-independent approach (aspect 1).

The authors mention the fact that in service provisioning of the public administration domain a large number of different authorities might be involved. The aspect of collaborative construction (aspect 3) is not covered by the proposed methodology, though.

Besides IT specialists also domain experts of public administration are identified as an important stakeholder group who are responsible for establishing a common understanding of the service interface, analyzing information demand and quality requirements as well as determining topics, terms and relations to be used. This methodology is directly intended for the public administration sector. Consequently it should also consider essential legal aspects of respective public administration domains. However, the methodology does not include any validation step where domain experts could ensure legal certainty. The actual ontology design is conducted by classical ontology engineers (aspect 5).

The authors do not mention any tool support for the methodology (aspect 4).

F. Methodology 6

Developing Ontology-Grounded Methods and Applications (DOGMA) [18] represents an ontology engineering methodology that is aimed at building both highly reusable and usable ontologies. Concerning aspect 1, this is the only methodology that covers both application-dependence as well as application-independence in one approach and highlights the importance of developing reusable as well as usable ontologies. This goal is reached by introducing a shared ontology base that consists of "plausible" domain axiomatizations and application axiomatizations. Application axiomatizations consist of a selected set of lexons from the ontology base and a specified set of rules to constrain the usability of these lexons [18].

Development is supported by the so-called DOGMA Studio Workbench (aspect 4) that also provides plugins for a community layer that aims at supporting the DOGMA-MESS methodology [19]. DOGMA-MESS emphasizes on providing guidelines for collaborative and interorganizational ontology engineering (aspect 3). Thereby, the authors discuss that “a viable methodology requires not building a single, monolithic domain ontology by a knowledge engineer, but supporting domain experts in gradually building a sequence of increasingly complex versions of interrelated ontologies over time”.

The process of ontology building is hierarchically structured. Every domain has a so-called Upper Common Ontology that is maintained by the core domain expert. The most important artifacts of this ontology are templates that describe a common knowledge definition. Over time, templates should become more numerous and should evolve during multiple iterations of development. Templates are then specialized into organizational specializations by the domain experts representing different organizations. The authors present a so-called Lower Common Ontology for negotiating the meaning of specific terms (aspect 2).

The authors refer to the importance of domain experts in interorganizational ontology engineering and also include human-centered aspects in the respective software tool (aspect 5).

G. Methodology 7

In contrast to other ontology engineering methodologies, the NeOn methodology [20] does not define a rigid process to follow, but instead, it suggests a variety of pathways for developing ontologies. It defines nine different scenarios, a glossary of processes and activities, two ontology life cycle models (waterfall life cycle model, iterative-incremental life cycle model) as well as a set of methodological guidelines for different processes and activities (aspect 2) [20].

The authors discuss the fact that due to the increase of online available ontologies ontology development is more and more becoming a reuse-centric process. Consequently, ontology development can be characterized as the construction of a network of ontologies, managed by different people and different organizations. Thus, the proposed methodology particularly aims at providing support for the collaborative construction of ontology networks (aspect 3) [20].

The methodology is intended for the classical ontology engineer who is defined as software developer or ontology practitioner involved in the development of ontologies. Hence, the methodology does not include any guidelines for non-experienced domain experts to autonomously develop ontologies. However, the methodology includes a well-elaborated evaluation activity, which could also incorporate safeguarding legal certainty (aspect 5).

The NeOn toolkit provides explicit support for developing ontologies following the proposed methodology (aspect 4).

As it is the aim of the authors to define a generic framework for the development of ontologies, it is completely application-independent (aspect 1).

IV. Results

As shown in Table I, it can be observed that most of the methodologies suggest a rather generic and application independent approach to ontology engineering. In contrast, Methodology 4 is developed for a specific domain (organizational learning) and system (APOSDLE) aiming at a rapid application of developed ontologies. Additionally, Methodology 6 discusses that both usability and reusability of ontologies are important. Hence, this methodology focuses on application independence as well as application dependence.

The recommended processes and life cycles range from classical waterfall development, to iterative and incremental development. In our opinion, in this context no approach can be seen as better than another. Whereas most of the presented methodologies recommend only one procedural model Methodology 7 defines several of them. The method describes use-cases that should help to identify the most appropriate process for a given situation.

Most of the investigated methodologies name collaborative construction of ontologies as an essential goal. Methodology 1, Methodology 2, Methodology 3, and Methodology 4 include explicit assistance aiming at structured conversations between all participants. Whereas the first two follow a rather centralized approach with a control board that manages inputs from participants, the other methodologies prefer a more interactive and agile approach resulting in faster response times.

TABLE I. SUMMARY OF STUDY RESULTS

Methodology	Asp. 1	Asp. 2	Asp. 3	Asp. 4	Asp. 5
1	Appl. ind.	Iterative	Yes	No	OE
2	Appl. ind.	Iterative	Yes	Wiki	OE/DE
3	Appl. ind.	Iterative	Yes	HCONE	DE
4	Appl. dep.	Agile	Yes	Moki	OE/DE
5	Appl. ind.	Waterfall	No	No	OE/DE
6	Appl. ind. Appl. dep.	Iterative	Yes	DOGMA Studio	OE/DE
7	Appl. ind.	Waterfall Iterative	Yes	NeOn Toolkit	OE

OE: Ontology Engineer; DE: Domain Expert

Many methodologies identify the domain expert as a crucial participant in the ontology engineering process. For example, Methodology 6 discusses that “… an interorganizational ontology needs to be modeled not by external knowledge engineers, but by domain experts themselves. Only they have the tacit knowledge about the domain and can sufficiently assess the real impact of the conceptualizations and derived collaborative services on their organization. …” However, it is interesting to observe that only Methodology 3, Methodology 4, and Methodology 6 offer explicit support for domain experts to model the respective ontologies, or at least parts of it, autonomously.

V. RELATED WORK

A very comprehensive comparative study that presents the most representative methodologies used in ontology development at that time was conducted by Fernandez-Lopez [6]. The study analyses methodologies against the IEEE Standard for Developing Software Life Cycle Processes (1074-1995). The author already mentions a criterion for collaborative and distributive construction but comes to the conclusion that none of the publications at that time cover this aspect explicitly.

A very similar study has been conducted by Fernández-López and Gómez-Pérez [21] that additionally introduces a methodology categorization. The categorization includes methodologies for building ontologies from scratch, methodologies for reengineering ontologies and methodologies for collaborative construction.

The study by Beck and Pinto [22] gives a rather informal overview of methodologies for ontologies. The paper emphasizes aspects like "consider reuse" and also mentions life cycles and typical ontology engineering activities (e.g., specification, conceptualization, formalization, implementation, or maintenance).

Corcho, Fernández-López, and Gómez-Pérez [23] additionally describe ontology tools and ontology languages available at that time. The authors come to the conclusion that future work should be driven towards the creation of a common workbench that supports ontology development during the whole life cycle, ontology management, ontology support as well as methodological support for building ontologies.

Sandkuhl [24] provides an analysis of ontology development methodologies in the context of small and medium-sized enterprises (SMEs). The study focuses on reducing development time for building ontologies. Thus, the study analyses aspects like completeness of the methodology, life cycle coverage and reuse of already existing ontologies.

Kim and Choi [25] present an evaluation of ontology development methodologies with CMM-i. Although the idea of taking CMM-i (a very comprehensive framework for organizations to assess their development and maintenance processes) as evaluation framework sounds promising the actual study does not present many valuable results.

Casellas [10] presents the latest approach in the field of comparing ontology engineering methodologies. The article can be seen as a recommendation in terms of listing most of the relevant methodologies currently available. However, the analysis tends to focus only on the followed life cycle of the studied methodologies and lacks in taking more analysis criteria into account.

VI. CONCLUSION

Considering aspects and requirements of the E-Government domain described in Section I and II, we come to the conclusion that none of the analyzed methodologies is fully mature to serve as a "domain expert centered ontology engineering methodology in the context of electronic service provisioning in public administration". In fact, there is one single candidate (i.e., Methodology 4) that technically addresses all methodological requirements (see aspects 3 to 5, Section II) in an acceptable way. Unfortunately, this methodology was developed for a different domain (i.e., organizational learning). Hence, its domain-dependence and application-dependence make a direct exploitation for the E-Government domain very difficult. However, aspects and general guidelines of this as well as of some other methodologies can definitely contribute to a future methodology in this field.

Methodology 7 suggests different activities and processes depending on a specific situation and does not follow a rigid workflow for every situation. In fact, a public administration subsumes a variety of different domains (e.g., welfare, health, buildings and constructions, education). This methodology may take into account that not each public domain, each public service or each modeling activity may fit into one single process model, as a potential advantage.

Methodology 6 proposes that not only reusability but also usability of ontologies in specific ontology-driven applications is important. This contributes to the situation that domain experts in general often lack in sufficient abstraction abilities. Hence, in order to be able to validate consistency and reasonableness of resulting ontologies domain experts should be able to check the consequences of modeling decisions in respective applications immediately.

As ontology engineering in public administration generally involves numerous experts from different agencies, expert knowledge is usually scattered over the involved participants. Enabling a structured conversation among all participants is crucial for an ontology engineering methodology in this context. On the one hand, this should lead to a collaborative construction of the domains in question and on the other hand should also assure the differentiation of responsibilities and roles. Aspects of Methodology 1, Methodology 2, Methodology 3, and Methodology 4 may valuably contribute to this requirement.

Many methodologies include a revision or validation activity in the proposed process. Whereas in an engineering-centered development approach validation activities usually deal with formal issues of the ontology a domain expert-centered approach should additionally emphasize factual aspects. For example, Methodology 2 asks questions like "does it represent the domain, or does it contain factual errors?" In the public administration sector, also legal aspects have to be considered in this respect. This circumstance is hardly ever addressed by the studied methodologies.

Methodology 4, Methodology 6 and Methodology 3 already include some general guidance for inexperienced ontology engineers. In this respect, the authors of [17] mention the problem that domain experts are often rarely available and scarcely motivated towards modeling. Consequently, we infer that firstly, much more effort is necessary to develop methods and tools that further reduce the complexity of ontology engineering, and secondly, future research has to pay special attention to improving the commitment of domain experts.

 ISBN: 978-1-61208-293-6

To conclude, one important implication of introducing semantic technologies to the E-Government sector is to increase transparency of the decision making process as well as to increase the citizen orientation of public agencies. Proposing an ontology engineering methodology in the context of public administration would definitely be a next step for an increased matureness of such semantic initiatives. The requirements-driven approach to ontology engineering as proposed by Methodology 5 that starts with the viewpoint and desires of the citizens who want to consume public services already addresses this issue and may therefore also contribute to a future methodology in this field.

REFERENCES

[1] V. Peristeras, K. Tarabanis, and S. K. Goudos, "Model-driven eGovernment interoperability: A review of the state of the art," Computer Standards & Interfaces, vol. 31, no. 4, Jun. 2009, pp. 613–628.

[2] R. Klischewski and S. Ukena, "Designing semantic e-Government services driven by user requirements," in Proceedings of ongoing research project contributions and workshops 6th International EGOV Conference Trauner Verlag Linz Austria, 2007, pp. 1–8.

[3] J. V. F. Dombeu and M. Huisman, "Combining Ontology Development Methodologies and Semantic Web Platforms for E-government Domain Ontology Development," International journal of Web & Semantic Technology, vol. 2, no. 2, 2011, p. 14.

[4] R. Klischewski and L. Lessa, "Sustainability of e-Government Success: an Integrated Research Agenda," in International Federation for information Processing (IFIP) e-Government Conference 2012, 2012, pp. 153–162.

[5] D. Liebwald, "Knowledge Representation and Modelling Legal Norms : The EU Services Directive," in Proceedings of the Third Workshop on Legal Ontologies and Artificial Intelligence Techniques, Jun. 2009.

[6] M. Fernandez-Lopez, "Overview Of Methodologies For Building Ontologies," in Proceedings of the IJCAI99 Workshop on Ontologies and ProblemSolving Methods Lessons Learned and Future Trends CEUR Publications, 1999, vol. 1999, pp. 1–13.

[7] H. Lutz and D. Stelzer, Informationsmanagement: Grundlagen, Aufgaben, Methoden (German), 2nd ed. Munich: Oldenbourg Wissenschaftsverlag, 2011.

[8] M. Jarrar and R. Meersman, "Ontology Engineering -The DOGMA Approach," in Advances in Web Semantics I, vol. Volume LNC, Springer, 2008, pp. 7–34.

[9] T. D'Entremont and M.-A. Storey, "Using a degree of interest model to facilitate ontology navigation," 2009 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), Sep. 2009, pp. 127–131.

[10] N. Casellas, "Methodologies, Tools and Languages for Ontology Design," in Legal Ontology Engineering - Methodologies, Modelling Trends, and the Ontology of Professional Judicial Knowledge, Springer, 2011, pp. 57–109.

[11] M. Uschold and M. King, "Towards a Methodology for Building Ontologies," in Proceedings of IJCAI95's Workshop on Basic Ontological Issues in Knowledge Sharing, 1995, vol. 82, no. 1, pp. 74–82.

[12] C. W. Holsapple and K. D. Joshi, "A collaborative approach to ontology design," Communications of the ACM, vol. 45, no. 2, 2002, pp. 42–47.

[13] D. Vrandecic, S. Pinto, C. Tempich, and Y. Sure, "The DILIGENT knowledge processes," Journal of Knowledge Management, vol. 9, no. 5, 2005, pp. 85–96.

[14] W. C. Mann and S. A. Thompson, "Rhetorical Structure Theory: A Theory of Text Organization," Ablex Publishing Corporation, 1987.

[15] C. Tempich, D. Vrandečić, and R. Benjamins, "OPJK modeling methodology," 2005.

[16] K. Kotis and G. A. Vouros, "Human-centered ontology engineering: The HCOME methodology," Knowledge and Information Systems, vol. 10, no. 1, 2005, pp. 109–131.

[17] C. Ghidini, M. Rospocher, B. Kump, V. Pammer, A. Faatz, and A. Zinnen, "Integrated Modelling Methodology - Version 2 . 0," 2009.

[18] P. Spyns, Y. Tang, and R. Meersman, "An ontology engineering methodology for DOGMA," Applied Ontology vol. 2, no. 3, 2008, pp. 13–39.

[19] A. De Moor, P. De Leenheer, and R. Meersman, "DOGMA-MESS : A Meaning Evolution Support System for Interorganizational Ontology Engineering," Engineering, vol. 4068, 2006, pp. 189–202.

[20] M. C. Suarez-Figueroa, A. Gomez-Perez, and M. Fernandez-Lopez, "The NeOn Methodology for Ontology Engineering," in Ontology Engineering in a Networked World, M. C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, and A. Gangemi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 9–34.

[21] M. Fernández-López and A. Gómez-Pérez, "Overview and analysis of methodologies for building ontologies," The Knowledge Engineering Review, vol. 17, no. 2, 2002, pp. 129–156.

[22] H. Beck and H. S. Pinto, "Overview of Approach , Methodologies , Standards , and Tools for Ontologies,". unpublished.

[23] O. Corcho, M. Fernández-López, and A. Gómez-Pérez, "Methodologies, tools and languages for building ontologies. Where is their meeting point?," Data & Knowledge Engineering, vol. 46, no. 1, 2003, pp. 41–64.

[24] K. Sandkuhl, "Towards a Methodology for Ontology Development in Small and Medium-Sized Enterprises," in IADIS Conference on Applied Computing, 2005, pp. 369–376.

[25] J. A. Kim and S. Y. Choi, "Evaluation of Ontology Development Methodology with CMM-i," in 5th ACIS International Conference on Software Engineering Research, Management & Applications (SERA 2007), 2007, pp. 823–827.

[26] Advanced Process- Oriented Self- Directed Learning Environment project site, http://www.aposdle.tugraz.at [retrieved: June, 2013].

[27] Web Service Modeling Ontology project site, http://www.wsmo.org [retrieved: June, 2013].

[28] OWL-S: Semantic Markup for Web Services W3C Member Submission, http://www.w3.org/Submission/OWL-S/ [retrieved: June, 2013].