

An Argumentation Ontology for DIstributed, Loosely-controlled and evolvInG Engineering processes of oNTologies (DILIGENT)

Christoph Tempich², H. Sofia Pinto¹, York Sure², and Steffen Staab³

¹Dep. de Engenharia Informática, Instituto Superior Técnico, Lisboa, Portugal
<http://www.dei.ist.utl.pt/>
sofia.pinto@dei.ist.utl.pt

²Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany
[http://www.aifb.uni-karlsruhe.de/WBS/](http://www.aifb.uni-karlsruhe.de/WBS/{sure,tempich}@aifb.uni-karlsruhe.de)
{sure,tempich}@aifb.uni-karlsruhe.de³ISWeb, University of Koblenz Landau,
56016 Koblenz, Germany
<http://www.uni-koblenz.de/FB4/>
staab@uni-koblenz.de

Abstract. A prerequisite to the success of the Semantic Web are shared ontologies which enable the seamless exchange of information between different parties. Engineering a shared ontology is a social process. Since its participants have slightly different views on the world, a harmonization effort requires discussing the resulting ontology. During the discussion, participants exchange arguments which may support or object to certain ontology engineering decisions. Experience from software engineering shows that tracking exchanged arguments can help users at a later stage to better understand the assumptions underlying the design decisions. Furthermore, as the constructed ontology becomes larger, ontology engineers might argue in a contradictory way without knowing so. In this paper we present an ontology which formalizes the main concepts which are used in an DILIGENT ontology engineering discussion and thus enables tracking arguments and allows for inconsistency detection. We provide an example which is drawn from experiments in an ontology engineering process to construct an ontology for knowledge management in our institute. Having constructed the ontology we also show how automated ontology learning algorithms could be taken as participants in the OE discussion. Hence, we enable the integration of manual, semi-automatic and automatic ontology creation approaches.

1 Introduction and Motivation

A prerequisite to the success of the Semantic Web are shared ontologies which enable the seamless exchange of information between different parties. The engineering of a shared ontology is a social process which (1) involves many participants – knowledge engineers, domain experts and users –, (2) may take place at many locations, (3) and is not a once-only process. However, currently available methodologies to support (*cf.* [1]) ontology engineering (OE) for the Semantic Web focus their attention mostly on the centralized development of static ontologies by knowledge engineers and a small

number of domain experts. In [2, 3] we propose the DILIGENT OE process which will eventually result in a well tested methodology, to support DIStributed, Loosely-controlled and evolvInG Engineering of oNTologies. We believe that this process model is better suited for the requirements on OE in the Semantic Web.

The application of the DILIGENT process model has shown in several case studies that exchange of arguments constitutes a major part in collaborative ontology building. In the SEKT project¹, for example, it was decided that a core upper ontology (PROTON) will be used by all participants. The ontology will be the basis for application development and information sharing. For practical reasons one participant initially provided an initial version, more than eight partners are now discussing the further evolution of the ontology. The initial version does not yet meet the requirements of all partners. Since the participants are distributed in Europe the discussion takes place informally via email. Although the discussion started only recently, it is already difficult for newcomers to enter it since (1) tracing the exchanged arguments is virtually impossible, due to the amount of mails already exchanged, and (2) only some participants know which parts have already been discussed. These problems will be even harder to cope with in a complete distributed environment like the Semantic Web.

This observation from our case study is inline with the experiences made in software and requirements engineering. There, extensions of the IBIS methodology[4] are used to capture design deliberations, thus make them traceable, and formal models have been developed to allow for structured queries on the arguments[5]. They have shown that formal argumentation models enhance traceability of design decision, help in conflict resolution, enhance reusability and facilitates the integration of new participants in the design process. Although, these models are very general, we have identified several requirements – further elaborated in section 5 – for argumentation support and its formalization which are unique for OE processes:

1. General argumentation models allow for all types of arguments and are very flexible. However, we have shown that a restricted set of arguments can facilitate OE processes [2], thus a formal model for OE should take this into account.
2. Within general argumentation models, inconsistencies in the discussion can not be easily detected, since arguers do not formalize their arguments. Ontologies are themselves formal models, thus inconsistencies should be considered during the discussion.
3. Ontology Engineering is often augmented with input from Ontology Learning[6]. No methodology provides an integrated view on manual and automatically created ontologies. An ontology learning algorithm can be seen as an agent providing arguments for design decisions. This should be regarded an integral part in a formalized argumentation model for OE.

In this paper we present an ontology formalized in OWL DL² based on the IBIS argumentation model. With this ontology one can formalize the argumentation taking place in OE processes (*cf.* req. 1), by instantiating the ontology. Note, that the ontology does not aim at formalizing the exchanged arguments logically. With an appropriate

¹ <http://www.sekt-project.com>

² <http://www.w3.org/TR/owl-ref/>

reasoner, inconsistencies in the argumentation can be detected (*cf.* req. 2). Our formal model can be adapted to different types of argumentation, namely arguments from algorithms can easily be integrated (*cf.* req.3).

In the following, we start by providing an extensive review of the state of the art in ontology engineering, argumentation visualization and argumentation structures (section 2). We then present the DILIGENT process template (section 3). Before we go into detail in the argumentation ontology (section 6), we analyze some use cases (section 4) for it and analyze the corresponding requirements (section 5). Finally we give an example drawn from an ontology engineering experiment and conclude.

2 Related work

The ideas just briefly introduced above and further elaborated in section 4, require research from different areas. In particular we looked into *ontology engineering*, *argumentation visualization*, *argumentation types* and *formal arguments*.

2.1 Ontology engineering

Established methodologies for ontology engineering summarized in [1], focus on the centralized development of static ontologies. **METHONTOLOGY** [1] and the **OTK methodology** [7] are good examples for this approach. They offer guidance for building ontologies either from scratch, reusing other ontologies as they are, or re-engineering them. They divide OE processes into several sub steps which produce an evaluated ontology for a specific domain.

Holsapple et al.[8] focus their methodology on the collaborative aspects of ontology engineering but still aim at a static ontology. A knowledge engineer defines an initial ontology which is extended and changed based on the feedback from a panel of domain experts. However, no support for argumentation is provided.

In [3] we show how OntoEdit was adopted to support **DILIGENT** OE processes (*cf.* section 3). Discussions are an important part of these kind of processes. However, no support for argument analysis is given.

A methodology which integrates argumentation and ontology engineering in a distributed setting is **HCOME** [9]. It supports the development of ontologies in a decentralized setting and allows for Ontology evolution. It introduces three different spaces in which ontologies can be stored: In the *Personal Space* users can create and merge ontologies, control ontology versions, map terms and word senses to concepts and consult the top ontology. The evolving personal ontologies can be shared in the *Shared Space*. The *Shared Space* can be accessed by all participants. In the shared space users can discuss ontological decisions based on the IBIS [10] model. After some discussion and agreement, the ontology is moved into the *Agreed space*. However, they do not present any experiences or adaptations to the IBIS model for OE.

This is the focus of the work described in [11]. A three-phased knowledge mediation procedure is proposed and evaluated. This approach is especially conceived to integrate different perspectives and information needs into one consensual ontology. They identify useful questions which can guide actors in an ontological discussion. However, they

do not analyze the dominant types of arguments which are used in these discussions. The main finding is that a moderator greatly enhances the efficiency and effectivity of the discussion.

In [4], a case study in building an ontology combining three existing ones is described. In this case study the **Compendium** tool was used to guide the discussion in a synchronous meeting. The results show that structured argumentation is beneficial for ontology engineering. The traceability of the decisions was enhanced. However, the authors were more concerned with the evaluation of their tool than with the specific issues arising in a discussion about an ontology. The authors do not examine which kinds of arguments are exchanged and how the discussion could be made more efficient.

2.2 Argumentation Visualization

As structured argumentation support was identified as beneficial for OE we here summarize the development of this field briefly. The most accepted model of argumentation is the **IBIS methodology** [10]. IBIS was developed to provide a simple yet formal structure for the discussion and exploration of “wicked” problems. Wicked problems cannot be solved in the traditional sense, because one runs out of resources (time, money, energy, people, etc.) before a perfect solution can be implemented.

gIBIS [12] focuses on capturing collaborative deliberations during design activities in the form of graphs containing text at their nodes. It was the first graphical interface for the IBIS methodology. IBIS allows to capture different design deliberations. Appropriate tools can later on help to retrieve them in a sophisticated way. For example the requirements engineering community has long identified the need to capture the arguments exchanged during the design process to enhance traceability. [5] presents an early formalization of the IBIS model. However, the IBIS methodology was criticized due to its resilience to change and for being too abstract. In [13] it is argued that IBIS should be enhanced with domain specific knowledge. The work reported in [14] further enhances the IBIS methodology by introducing an acceptance and rejection mechanism. They emphasize the restriction to record only important considerations.

Compendium[15] builds on the gIBIS system. It is a semantic hypertext tool to capture arguments and visualize them. It offers a conceptual framework for argumentation, it promotes the use of a meeting facilitator and it includes tools to present the exchanged arguments customized for audiences. Compendium tools include *Question based templates* to ease the flow of the arguments. Hence, the discussion can be lead by pre-formulated questions which structure the discussion. A discussion is visualized by different maps, interlinking and connecting the exchanged arguments. Any idea can be expressed in Compendium since its notation is very flexible.

2.3 Types of arguments

Argumentation models provide a conceptual model for the interaction of issues, ideas and arguments. However, they do not differentiate the different kinds of arguments arising in a discussion. In [2] we analyzed OE discussions with the help of Rhetorical

Structure Theory (RST) [16].³ In our experiments the actors in the OE discussions had the task to agree on a shared ontology to represent the research topics of our institute. We conducted two experiments. In the first the actors were free to discuss the ontology with little guidance, whereas in the second we restricted the types of arguments allowed in the discussion to the ones more relevant and effective to reach consensus identified in the first experiment. The result was that restricting the argument types to Elaboration, Evaluation, Justification, Contrast, Alternative, Example and Counter Example enhances the productivity of OE discussions. Furthermore, the actors kept better track of the discussion, the agreement process was facilitated, and agreement was reached faster.

2.4 Formal arguments

Formalization of arguments is an important topic in the AI community. Although OWL provides us a formalism that allows to formally state arguments, we do not believe that ontological decisions can be discussed in a completely formal way, at least if the ontology is to be used by humans. Several results show the advantages of using formal models. For example [17] proposes a formal model of argumentation, using the IBIS argumentation model. Based on the formal arguments a preferable solution can be derived. Another interesting application is argument selection based on user needs. For example [18] presents how formal argumentation trees can be pruned to best correspond to users wishes. However, this is not the focus of our work.

2.5 Summary

Review of existing OE methodologies reveals that there is no completely elaborated methodology integrating collaborative, distributed and evolutionary aspects. Moreover, none currently supports the combination of manual and semi-automatic OE approaches. The use of argumentation methodologies for OE was recently recognized but is not yet concisely integrated into the OE process. A formal argumentation model to assist OE is completely missing.

3 DILIGENT process

In order to provide enough background knowledge about the DILIGENT argumentation ontology, which we present in this paper, we here sketch the overall framework, in which it is embedded, i.e. the overall DILIGENT process (cf. [3]).

Scenario In *distributed* development there are several experts, with different and complementary skills, involved in collaboratively building the same ontology. For instance, in Virtual Organizations, Open Source and Standardization efforts, experts belong to

³ RST originally offers an explanation of the coherence of texts. It is assumed that for every part of a coherent text there is some function, thus has a particular argument type. Thirty different arguments types have already been identified and loosely defined.

different competing organizations and are geographically dispersed. In these cases, builders typically are also users and, although some users are not directly involved in changing the ontology, they take part in the process by using the ontology.

Process We will now describe the general process, roles and functions in the DILIGENT process. It comprises five main activities: (1) **build**, (2) **local adaptation**, (3) **analysis**, (4) **revision**, (5) **local update** (cf. figure 1). The process starts by having *domain experts, users, knowledge engineers* and *ontology engineers* **building** an initial ontology. This can be supported by using ontology learning tools. In contrast to known ontology engineering methodologies available in the literature [1, 19, 20] our focus is distributed ontology development involving different stakeholders, who have different purposes and needs and who usually are not at the same location. Therefore, they require online ontology engineering support. The team involved in building the initial ontology should be relatively small, in order to more easily find a small and consensual first version of the shared ontology. Moreover, we do not require completeness of the initial shared ontology with respect to the domain.

Once the product is made available, users can start using it and **locally adapting** it for their own purposes. Typically, due to new business requirements, or user and organization changes, their local ontologies evolve in a similar way as folder hierarchies in a file system. In their local environment they are free to change the reused shared ontology. However, they are not allowed to directly change the ontology shared by all users. Furthermore, the control board collects change requests to the shared ontology.

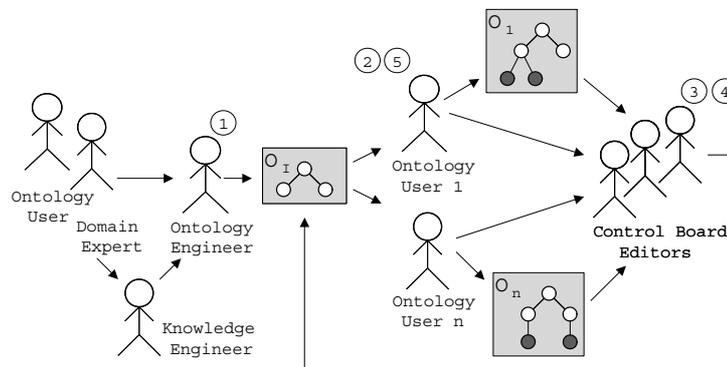


Fig. 1. Roles and functions in distributed ontology engineering

The board **analyzes** the local ontologies and the requests and tries to identify similarities in users' ontologies. Since not all of the changes introduced or requested by the users will be introduced,⁴ a crucial activity of the board is deciding which changes are going to be introduced in the next version of the shared ontology. The input from users provides the necessary arguments to underline change requests. A balanced decision that takes into account the different needs of the users and meets user's evolving

⁴ The idea in this kind of development is not to merge all user ontologies.

requirements⁵ has to be found. The board should regularly **revise** the shared ontology, so that local ontologies do not diverge too far from the shared ontology. Therefore, the board should have a well-balanced and representative participation of the different kinds of participants involved in the process.

Once a new version of the shared ontology is released, users can **update** their own **local** ontologies to better use the knowledge represented in the new version. Even if the differences are small, users may rather reuse *e.g.* the new concepts instead of using their previously locally defined concepts that correspond to the new concepts represented in the new version.

We have applied this process model to the case of folder sharing via a Peer-to-Peer setting with centralized core folder structures and individual specific folder structures. Our experiences there have substantiated the validity of DILIGENT (cf. [3]).

Threads of arguments A central issue in the DILIGENT process is keeping track of threads of exchanged arguments. We can identify several stages in which arguments play an essential part:

- Ontology is defined as “a shared specification of a conceptualization” [21]. Although “shared” is an essential feature, it is often neglected. In DILIGENT experts exchange arguments while **building** the initial shared ontology in order to reach consensus;
- When users make comments and suggestions to the control board, based on their **local adaptations**, they are requested to provide the arguments supporting them;
- while the control board **analyzes** the changes introduced and requested by users, and balances the different possibilities, arguments are exchanged and balanced to decide how the shared ontology should change.

4 Use Case

As mentioned in the introduction the SEKT-project partners are currently building a common upper ontology (PROTON⁶). PROTON will be used in the applications developed in SEKT as background knowledge. Hence, PROTON will be used in case studies tackling knowledge management in a telecom company and a question answering system for legal education. Furthermore, PROTON serves as background knowledge for natural language processing and machine learning methods. Naturally, the applications have different requirements on the ontology. However, to maximize interchangeability of methods developed in SEKT, it was agreed to build a common ontology. From a DILIGENT process point of view, we are currently in the revision phase. Some of the partners have already used a previous version of PROTON and adapted it according to their needs⁷. Others have just recently joined the process. The discussion takes place on a mailing list. In the following we first summarize the experiences made so far and then introduce some additional features needed for the DILIGENT *Argumentation Ontology*.

⁵ This is actually one of the trends in modern software engineering methodologies (see Rational Unified Process).

⁶ <http://proton.semanticweb.org/>

⁷ The name was agreed only recently.

4.1 Traceability

As new partners get involved into the ontology building process, modelling decision are discussed more than once, since the modelling reasons of the existing version are not documented. Although, the actors in the current OE discussion present reasons for design modification, the number of e-mails makes it infeasible to retrieve them at a later stage. The ability to present the reasons and arguments for a modelling decision to the **new entrants** could speed up the design process. A similar problem arises, when the ontology is **revised** and the ontology engineers need to recall the reasons for the previous design. The users of the ontology can as well profit from a well documented ontology for a better understanding. Currently, they rely on the sparse explicit documentation, since documentation is a time consuming, often neglected task. A structured integration of the ongoing discussions can ease it.

Another issue is size. The current version of PROTON has more than two hundred concepts. Therefore, it is difficult to track which parts of the ontology are agreed and which are not. In an OE discussion actors often agree only implicitly with a certain modelling decision. For example a participant proposes B as subconcept of A without explicitly agreeing with A.

Besides the current experiences, first versions of PROTON had to be build from scratch. Although there are a number of ontologies available on the Semantic Web, this is not sufficient for an ontology to be reused. Only if the design rationales behind the model are available to others, can ontologies easily be included into applications.

4.2 Inconsistency detection

During the argumentation process different participants exchange their opinions about the issue under discussion. A requirement on an efficient discussion is, that the arguments one participant brings forward are consistent with his previous arguments. A participant may change his opinion, but then he should discard earlier contradicting arguments. A model to conceptualize arguments should be able to detect at least some inconsistencies and point the arguer to the contradicting arguments 1.

Inconsistency	Description
Idea inconsistency	Arguer introduces Idea1 and Idea2 which are inconsistent
Argumentation inconsistency	Arguer argues first in favor and then against an issue. The lines of reasoning followed by the arguer lead to inconsistent ideas
Position inconsistency	Assuming Issue/Argument 1 and 2 are contradicting. An Actor produces a position inconsistency when he votes in favor of Issue/Argument 1 and then introduces Issue/Argument 2

Table 1. List of possible inconsistencies

4.3 Argument selection

In the applications for the SEKT case studies the user of PROTON may wonder why some concepts, etc. were introduced in the ontology or he may ask why certain modelling decisions were made. However, even when we trace the underlying arguments, some of them may be very detailed and not understandable to normal users. Hence, if a user asks for the arguments underlying the ontology modelling decisions it would be beneficial to provide an answer which best fits the users needs. In this case we can assume that the best answer to such a query would be one which convinces the requester most. The selection of the appropriate arguments is only possible if not only the argumentation but also the arguments are formalized. Then we can build on models as presented in [18] that show how formal argumentation trees can be pruned to best correspond to the users wishes. On the other hand in tangled discussion it is not always obvious which proposal receives the strongest support. [17] presents a formal model to establish the winner of a discussion.

5 Requirements

We have identified several requirements for our *Argumentation Ontology* from the the SEKT PROTON case study and others where we have been involved such as IEEE SUO. Before we describe the ontology in the next section we now develop its requirements for it.

1. **Use common vocabulary** Research in argumentation and its visualization has a long history and is a mature field (*cf.* 2). To enhance acceptability for the ontology usage of the established vocabulary is essential.
2. **Focus on relevant arguments** As observed in [2] the restriction of available argument types can focus and speed up OE discussions. Hence the ontology should not model all possible kinds of arguments of a discussion, but focus on the relevant ones. This view is supported by [22] who have developed an ontology for a different domain but for a similar purpose and found that a smaller ontology enhances usability.
3. **Ontology focus** Following the results of [13], that IBIS should be enhanced with domain specific knowledge, the developed ontology should be particularly well suited for ontology design.
4. **Adaptivity** The *Argumentation Ontology* should allow for capturing the structure of argumentation. Hence, the design must take into account that e.g. humans discuss on a free text basis while ontology learning algorithms use formal, structured and detailed reasons for different proposals.
5. **Support entire argumentation** The *Argumentation Ontology* should support the full argumentation cycle. This includes issue raising, conflict mediation, bargaining, clarification and agreement. Participants should be aware of which issues are currently under discussion, postponed, agreed and discarded.
6. **Conceptual as well as formalization level** People might agree on the need for a certain conceptual model but not on its actual implementation. The model should support argumentation on both conceptual and formal models.

7. **Modularization** Although the ontology should support the ontology engineering process we do not aim to support every part of it. As described in [7] the ontology engineering process involves the definition of requirements, owners and other meta attributes like Dublin core meta data. These should not be modelled here.
8. **Formalism independence** The *Argumentation Ontology* should be independent of the formalism used to model the final ontology. Each formalism allows different sets of modelling decisions and all can be subject to discussion. However, the formal model of the finally agreed ontology should be a result of the instantiation of the *Argumentation Ontology*.
9. **Process awareness** The *Argumentation Ontology* is embedded into the DILIGENT process presented in section 3. Essential properties of this process are its collaborative aspects, its distributiveness and the asynchronous way participants can provide arguments.
10. **Argumentation formalization** Although we do not currently plan to provide the arguments themselves in a formal way, the *Argumentation Ontology* should allow us to do so. As our last use case has illustrated ontology engineering and ontology usage could gain from such a formalization.

6 An Argumentation Ontology for DILIGENT processes

The DILIGENT *Argumentation Ontology* is visualized in figure 2⁸. The main concepts in our ontology are **issues**, **ideas** and **arguments**, which are represented as classes. These are in line with the terminology proposed by the IBIS methodology (req. 1). Issues introduce new topics in the discussion from a conceptual point of view. They are used to discuss what should be in the conceptual model of the ontology without taking into account how these items should actually be formalized and implemented in the ontology (req. 8). Ideas refer to how these concepts should be formally represented in the ontology, for instance as a class, an instance, etc. They relate to concrete ontology change operations⁹. Ideas are related to issues in the sense that they *respond to* them. Ideas refer to how issues should actually be implemented in the ontology. In this way discussions can take place in both the conceptual level and the formalization level (req. 6). Arguments are *arguments on* either one particular idea or one particular issue. Typically, our domain experts will start by proposing new issues to be introduced in the ontology. Arguments will be exchanged over them. Then, they discuss how these issues should be formalized through concrete ideas. Domain experts can also provide **elaborations**. These are issues that refine an issue under discussion, *elaborates on*.

Since concepts to be represented in an ontology should be consensual, this requires some consensus building discussions. In DILIGENT processes, concepts are only added to the ontology if they can be agreed upon, that is after some arguments have been exchanged, positions by different actors have been issued on them and some decisions have been made. Arguments for (pro) an idea or issue are called **justifications**. Arguments against (con) an idea or issue are called **challenges**. In what regards arguments in favor, particularly useful OE processes, we identified **examples** and **evalu-**

⁸ The corresponding OWL ontology will be available online in case of acceptance.

⁹ For example [23] presents a formal model for ontology change operations.

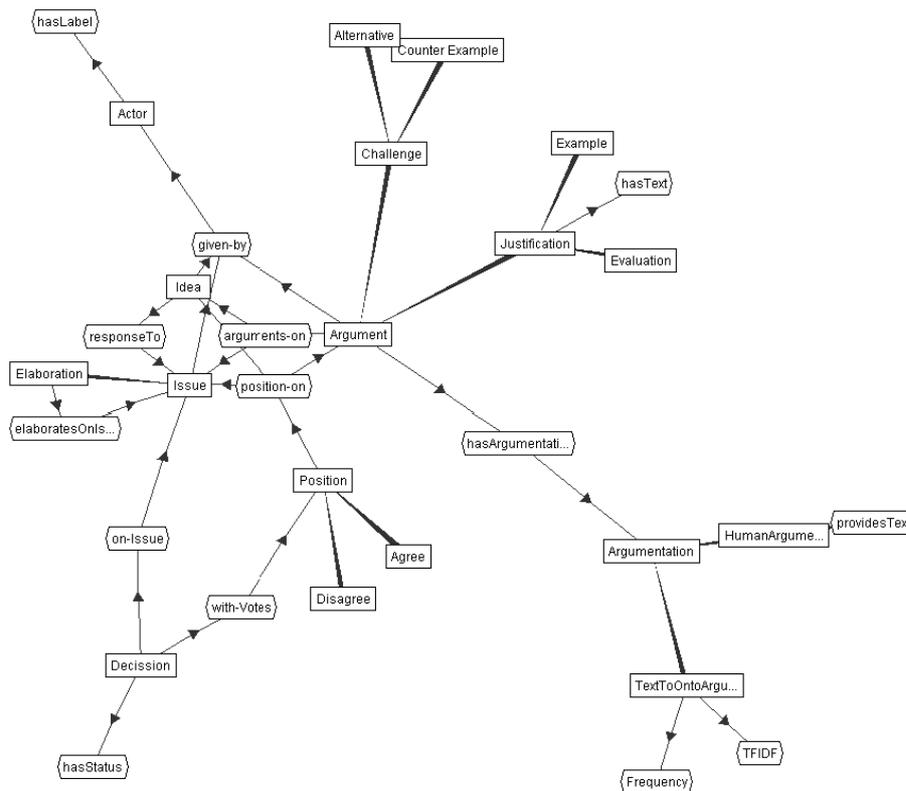


Fig. 2. The major concepts of the argumentation ontology and their relations

ation&justification. Two classes in challenges are also particularly used in OE discussions: **counter examples** and **alternative&contrast**. These arguments focus the IBIS argumentation methodology for Ontology Engineering (req 3).

Those involved in discussions can state **positions**. They clarify the *position on* one issue, one idea, or an argument under discussion. Either one **agrees** or **disagrees**. Once enough arguments have been provided and positions have been stated on them **decisions** can be made. In general, positions lead to **decisions**. Decisions are taken on issues. A decision has a status that can vary from **under-discussion**, **postponed**, **discarded** and **agreed** (req 5). A decision records not only the *issue on* which it was taken, but also both the positions issued when final *with-votes* (several positions) were cast and the *line of reasoning* (a sequence of arguments) underlying the decision on that issue. A decision can also state the *idea on-idea* underlying its issue. This allows one to focus on the relevant arguments (req 2).

Arguments are *given by* actors (req 9). We can have different kinds of Actors: either **Humans** or **Machines**. Different kinds of actors provide different argumentations (req. 4). In what regards argumentation humans (**HumanArgumentation**) tend to argue by providing strings of text stating (**provides text**) their reasons while machines

tend to use other kinds of argumentation measures like Frequency and TFIDF [6]. For each algorithm used, new subclasses of argumentation need to be introduced to model the different kinds of measures.

7 Example: An Argumentation Ontology for DILIGENT processes

The following discussion transcript was a part of an experiment performed at our institute (*cf.* [2], section 2.3). The participants were asked to build an ontology for modelling the research interests of our group. The experiment lasted for 90 min. and involved eleven actors. The participants provided their arguments in free text without formal restrictions. Hence, in the following example we model the discussion *ex post*. Moreover, we do not aim to model the entire discussion, but pick out an excerpt to exemplify our model.

...

cs: We have done quite a bit of research in distributed knowledge management (DKM) lately. So I suggest DKM as a topic plus a subtopic “peer to peer” (P2P)

The actor suggests on the one hand to introduce “DKM” and “P2P” in the ontology (Issues), and proposes on the other hand to model them as “topics” (Ideas).

Formalization

```
Individual(issue1 type(Issue) value(states “I suggest DKM”))
Individual(issue1 type(Issue) value(given-by actorCS))
Individual(justi1 type(Justification) value(hasArgumentation argumentation1))
Individual(justi1 type(Justification) value(arguments-on issue1))
Individual(argumentation1 type(HumanArgumentation) value(providesText “We
have ... lately”))
Individual(idea1 type(Idea) value(respondsTo issue1))
Individual(idea1 type(Idea) value(ontoChange add(DKM:Topic)))
Individual(elaboration2 type(Elaboration) value(states “P2P subtopic DKM”))
Individual(idea2 type(Idea) value(respondsTo elaboration2))
Individual(idea2 type(Idea) value(ontoChange add(DKM supertopic P2P)))
```

ah: I suggest knowledge management (KM) as super concept of DKM because every DKM is a kind of KM

The second actor agrees implicitly with the suggestion to introduce “DKM” in the ontology. In contrast to the first one he proposes to model it as a “concept”.

Formalization

...

```
Individual(idea3 type(Idea) value(ontoChange add(KM:Concept)))
```

...

jt: Well I am now wondering whether P2P is DKM, because File exchange is not always KM is it?

A third actor agrees also implicitly, that “P2P” and “DKM” are important for the domain, but challenges that they should be modelled in the proposed way.

Formalization

Individual(counter1 type(CounterExample) value(hasArgumentation argumentation3))

Individual(counter1 type(CounterExample) value(arguments-on elaboration2))

Individual(argumentation2 type(HumanArgumentation) value(providesText “File exchange ... KM”))

ph: I suggest Distributed Comp. (DC) with P2P and Grid as subtopics; DKM as subtopic of DC and KM

The fourth actor presents a new issues which could resolve the conflict.

Formalization

...

Individual(issue2 type(Issue) value(states “I suggest DC”))

Individual(elaboration3 type(Elaboration) value(... ..))

...

do: PRO *ph* : because his approach separates KM and distributiveness

The actor “do” agrees with the suggestion and provides additional reasons for the design. Implicitly he also agrees that “KM” should be part of the ontology.

Formalization

Individual(position1 type(Agree) value(position-on elaboration3))

...

cs: I’d like to agree to *ph* and *do* suggestion.

...

The first actor agrees with the new solution and discards his original proposal.

This example demonstrates that OE discussion can be modelled with the DILIGENT *Argumentation Ontology*. The applicability of the ontology will depend on the available tool support. We do not intent to automatically annotate a free discussion. We rather envision a template based approach. Currently we use a WIKI to support the argumentation process. However, integration with reasoners and inclusion into existing OE environments is desirable, but remains to be done.

8 Conclusion

It is now widely agreed that ontologies are a core enabler for the Semantic Web vision. The development of ontologies in centralized settings is well studied and there are established methodologies. However, current experiences from projects suggest, that ontology engineering should be subject to continuous improvement rather than a one-time effort and that ontologies promise the most benefits in decentralized rather than centralized systems.

In such settings, arguments play a major role in the process of consensus building between the involved participants. Based on the current state of the art in Ontology Engineering, Argumentation visualization and Argumentation structures, we propose an integrated formal argumentation model to be used in OE discussions, in particular in DILIGENT OE processes. This ontology supports the process in several ways. In discussions, it focuses the participants and helps to structure their arguments. In the usage and analysis phases, the exchanged arguments can be consulted to better understand the current version of the model. Moreover, since it is formal it allows for inconsistency detection in argumentations. Since the ontology covers all aspects of the discussion activity, namely issue raising, formalization of the issues and decision making, the participants are always informed about the current status of the discussion and the ontology they are building.

We demonstrate the applicability of our model by formalizing an OE discussion drawn from an experiment in our institute. The DILIGENT Argumentation Ontology will also be the basis for ontology discussions in the SEKT project. To support the discussion with appropriate tools we are currently investigating a combination of WIKI like argumentation support with ontology formalization in the KAON tool suit¹⁰. Argument selection based on formal arguments remains future work. In the further future we imagine that ontology learning methods can profit from the formalized discussion and learn from human ontology design decisions.

Therefore, the main contribution of this paper is the first formal argumentation model for Ontology Engineering, in particular for DILIGENT OE processes. This model is an adaptation of the IBIS argumentation model specifically for Ontology Engineering. It clearly distinguishes between phases: discussions should be about the conceptual model, about Issues, and about the formal model, about Ideas. Moreover, from our previous experiences in DILIGENT OE processes this model clearly states the arguments that have been identified as speeding and easing the consensus building process needed to build shared ontologies. Finally, this is the first model that attempts to integrate arguments from (semi-)automatic ontology building based on learning.

Acknowledgements. Research reported in this paper has been financed by EU in the the IST project SEKT (IST-2003-506826).

References

1. Gómez-Pérez, A., Fernández-López, M., Corcho, O.: Ontological Engineering. Advanced Information and Knowledge Processing. Springer (2003)
2. Pinto, H.S., Tempich, C., Staab, S., Sure, Y.: Diligent: Towards a fine-grained methodology for distributed, loosely-controlled and evolving engineering of ontologies. In de Mántaras, R.L., Saitta, L., eds.: Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004), August 22nd - 27th, Valencia, Spain, IOS Press (2004) 393–397
3. Pinto, H.S., Staab, S., Sure, Y., Tempich, C.: OntoEdit Empowering SWAP: a Case Study in Supporting Distributed, Loosely-Controlled and evolving Engineering of ontologies (DILIGENT). In Bussler, C., et al., eds.: Proceedings of the 1st ESWS 2004. (2004)

¹⁰ <http://kaon2.semanticweb.org>

4. Buckingham Shum, S., Motta, E., Domingue, J.: Augmenting design deliberation with compendium: The case of collaborative ontology design. In: HypACoM 2002: Facilitating Hypertext-Augmented Collaborative Modeling. ACM Hypertext'02 Workshop, University Maryland, MD (2002) Retrieved November 24, 2004 from <http://kmi.open.ac.uk/projects/compendium/SBS-HT02-Compendium.html>.
5. Ramesh, B., Dhar, V.: Supporting systems development by capturing deliberations during requirements engineering. *IEEE Trans. Softw. Eng.* **18** (1992) 498–510
6. Maedche, A., Staab, S.: Ontology learning for the semantic web. *IEEE Intelligent Systems* **16** (2001)
7. Sure, Y., Studer, R.: On-To-Knowledge methodology. In Davies, J., et al., eds.: *On-To-Knowledge: Semantic Web enabled Knowledge Management*. J. Wiley and Sons (2002)
8. Holsapple, C.W., Joshi, K.D.: A collaborative approach to ontology design. *Commun. ACM* **45** (2002) 42–47
9. Kotis, K., Vouros, G.A., Alonso, J.P.: HCOME: tool-supported methodology for collaboratively devising living ontologies. In: *SWDB'04: Second International Workshop on Semantic Web and Databases 29-30 August 2004 Co-located with VLDB*, Springer-Verlag (2004)
10. Kunz, W., Rittel, H.W.J.: Issues as elements of information systems. Working Paper 131, Institute of Urban and Regional Development, University of California (1970)
11. Aschoff, F.R., Schmalhofer, F., van Elst, L.: Knowledge mediation: A procedure for the cooperative construction of domain ontologies. In Abecker, A., van Elst, L., Dignum, V., eds.: *Proceedings of Workshop on Agent-Mediated Knowledge Management at the 16th European Conference on Artificial Intelligence (ECAI'2004)*, Valencia, Spain (2004) 20–28
12. Conklin, J., Begeman, M.L.: *gibis: a hypertext tool for exploratory policy discussion*. In: *Proc. of the 1988 ACM conference on Computer-supported cooperative work*. (1988)
13. Potts, C., Bruns, G.: Recording the reasons for design decisions. In: *Proceedings of the 10th international conference on Software engineering*, IEEE Computer Society Press (1988)
14. Gotel, O., Finkelstein, A.: Extended requirements traceability: Results of an industrial case study. In: *Proceedings of the 3rd IEEE International Symposium on Requirements Engineering (RE'97)*, IEEE Computer Society (1997) 169
15. Selvin, A., Buckingham Shum, S., Sierhuis, M., Conklin, J., Zimmermann, B., Palus, C., Drath, W., Horth, D., Domingue, J., Motta, E., Li, G.: *Compendium: Making meetings into knowledge events*. In: *Knowledge Technologies*, Austin, TX (2001)
16. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: A theory of text organization. In Polanyi, L., ed.: *The Structure of Discourse*. Ablex Publishing Corp., Norwood, N.J. (1987)
17. Gordon, T.F., Karacapilidis, N.: The zeno argumentation framework. In: *Proceedings of the sixth international conference on Artificial intelligence and law*, ACM Press (1997) 10–18
18. Hunter, A.: Towards higher impact argumentation. In McGuinness, D.L., Ferguson, G., eds.: *AAAI2004*, AAAI Press / The MIT Press (2004) 275–280
19. Pinto, H.S., Martins, J.: A Methodology for Ontology Integration. In: *Proc. of the First Int. Conf. on Knowledge Capture (K-CAP2001)*, New York, ACM Press (2001) 131–138
20. Uschold, M., King, M.: Towards a methodology for building ontologies. In: *Proc. of IJCAI95 WS*, Montreal, Canada (1995)
21. Gruber, T.R.: Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In Guarino, N., Poli, R., eds.: *Formal Ontol. in Conc. Analysis and Knowl. Rep.*, Kluwer Acad. Pub. (1993)
22. Buckingham Shum, S., Gangmin Li, V.U., Domingue, J., Motta, E.: Visualizing internet-worked argumentation. In Kirschner, P.A., et al., eds.: *Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making*. Springer (2003) 185–204
23. Stojanovic, L., Maedche, A., Motik, B., Stojanovic, N.: User-driven ontology evolution management. In: *Proceedings of the 13th European Conference on Knowledge Engineering and Knowledge Management EKAW*, Madrid, Spain (2002)