7th International
Command and Control Research and Technology Symposium

September 16 - 20, 2002
Québec City, QC, Canada

****

FINAL PAPER

****

1.1.1 ONTOCINC Server: A Web-based Environment for Collaborative Construction of Ontologies

Anne-Claire Boury-Brisset, Marlène Gauvin

Defence R&D Canada - Valcartier
2459 Pie-XI North, Val-Belair, QC, G3J 1X5, Canada
Phone: 1 (418) 844 4000
Fax: 1 (418) 844 4538

Email: {Anne-Claire.Boury-Brisset,Marlene.Gauvin}@drdc-rddc.gc.ca

Point of contact: Anne-Claire Boury-Brisset

Topic: Interoperability, standards

OntoCINC Server: A Web-based Environment for Collaborative Construction of Ontologies

Anne-Claire Boury-Brisset, Marlène Gauvin
Defence R&D Canada - Valcartier
2459 Pie-XI North, Val-Belair, QC, G3J 1X5, Canada
Phone: 1 (418) 844 4000, Fax: 1 (418) 844 4538
Email: {Anne-Claire.Boury-Brisset,Marlene.Gauvin}@drdc-rddc.gc.ca

Abstract

Coalition operations will become more and more important in the future. Interactions between coalition participants require mechanisms to facilitate the exchange of information. In this context, it is necessary to address interoperability issues so that coalition information can be effectively shared and exploited. In our research, we consider ontologies as a key component to provide a shared understanding of a domain and facilitate knowledge level interoperability among heterogeneous information sources. In this paper, we describe the specific requirements of coalition operations for information exchange and the methodology we proposed to the collaborative development of ontologies to satisfy these requirements. Then, we present the ontology engineering web-based environment we designed and implemented, called the OntoCINC Server. Finally, we present the lessons learned in applying the methodology within a coalition initiative.

2. **Introduction**

In the Canadian Forces, worldwide contributions are mainly performed through coalitions. Such operations will become more and more important in the future. Therefore, it will be necessary to provide Commanders with access to timely and relevant information from diverse and heterogeneous information sources for conducting their operations. To deal with this heterogeneity, future command and control information systems will have to address interoperability issues so that coalition information can be effectively shared and exploited. The problem of interoperability between systems is more challenging across organizations with different national doctrines. For example, one problem is that organizations often refer to the same concept using different names. To this end, interactions between coalition participants require mechanisms to facilitate the exchange of information and to provide a shared understanding of the domain based, at least, on a commonly agreed terminology. In this context, we consider ontologies as a key component to provide a shared understanding of a domain and facilitate knowledge level interoperability among heterogeneous information sources [Boury-Brisset, 2001]. An ontology formally defines a common set of terms that are used to describe and represent a domain, or to paraphrase T. Gruber [Gruber, 93], it is a formal explicit specification of a shared conceptualization.

The Defence Research & Development – Valcartier (DRDC – Valcartier) participates in a coalition experimental initiative, called C-CINC21 (Coalition - Commander in Chief of the 21st Century), which aims to define and conduct a set of multinational coalition command and control related experiments to advance the state of knowledge and to contribute to the interoperability of future coalition operations. Canada is one of the four-eye nations involved in this coalition initiative. Among the proposed activities, we aimed at experimenting with innovative knowledge management concepts and tools as well as to develop ontologies for coalition interoperability. In particular, one of the objectives of the C-CINC21 collaboration initiative is to provide an infrastructure and various services for information management within coalition environments built on top of ontologies, in order to help participants in the coalition to get improved situational awareness. The information services that could be provided include: semantic search and information retrieval among heterogeneous information sources based on shared ontologies, publishing of information, advertising of new publications, etc., as illustrated in Figure 1.

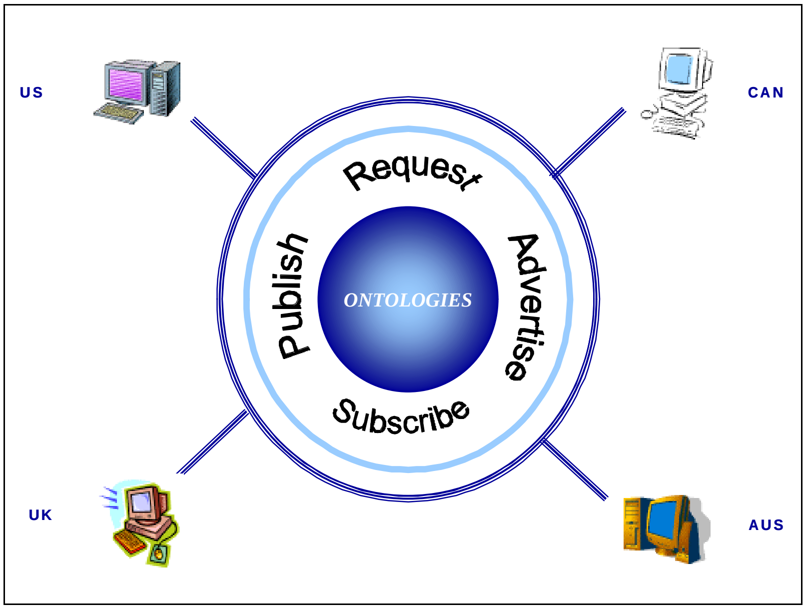

Figure 1. Information exchange services

In this context, we have initiated research activities to provide both a methodology for the development of ontologies and a tool facilitating the collaborative construction of ontologies. In this paper, we first describe the specific requirements of coalition operations for information exchange and the methodology we proposed to collaborative development of ontologies to satisfy these requirements. Then, we present the web-based environment we designed and implemented, called OntoCINC server. We show its functionalities and strengths to support the collaborative development of coalition ontologies and compare it to existing ontology development tools. We conclude the paper with the lessons learned from this experiment.

3. Proposed Methodology to Collaborative Ontology Development

Several ontology development methods have been proposed for a few years. A synthesis and comparison of such methods can be found in [Jones *et al.*, 1998]. However, it is becoming increasingly common for ontologies to be developed in distributed environments by authors with disparate background. Therefore, protocols for distributed ontology generation and maintenance are required. Some guidelines were recently proposed in response of these requirements ([Mc Guinness, 2000][Holsapple and Joshi 2002]). Ontology construction in a coalition context should be a collaborative and iterative process. One of the challenges is to bring people with disparate background and culture to collaboratively come out with a shared understanding of a domain. Another problem is to choose a representation formalism (meta-model) so that the ontology can be built upon it, validated with domain expert people (military people in our case) and exploited by computer systems. A flexible solution is needed.

The strategy is to exploit the richness of the diversity of experience and expertise instead of toning it down, and to follow a rigorous methodology. For this, we are adopting a high-level protocol for the collaborative building of ontologies. This includes stages where participants are involved as a team and stages where participants work individually.

The first stage consists of identifying a set of domain areas that are representative of the whole "universe" on which stands the collaborative initiative. In our context, domain areas were selected to support coalition operations, as illustrated in Figure 2.

Figure 2. Ontology domains

For each area, a leader is responsible for the construction of the ontology based on his/her experience related to the topic and works in collaboration with participants from other countries. This aims to maximize knowledge reusing in order to avoid building ontologies from scratch when there exists some information sources available or some work already done in another context. This is a recommendation from most of ontology construction methodologies, given that ontology building is a time-demanding activity. In addition to the business domain ontologies, support ontologies are also identified in order to represent temporal and spatial information as well as heterogeneous resources.

In a next stage, participants should agree on the formalism required to build and exchange the ontologies and eventually on a tool that would facilitate ontology construction. Ideally, the formalism should be kept at a conceptual level to facilitate (1) communication between people and (2) subsequent validation by domain experts. If required, the specification of the ontology meta-model is performed at this stage.

To summarize, the guidelines for the collaborative tasks to be achieved prior to ontology construction are the following:

- Identify domain areas to be captured in ontologies. This includes:
  - business ontologies and
  - support ontologies (representation of time/space, resource, etc);
- Identify a leader/nation responsible for each identified ontology;
- Determine the formalism and the engineering tool that will be used to capture and exchange the ontology; and if required,
- Specify the meta-model from which is ontology will be represented.

The development of the ontologies is the next stage. Even if this is a collaborative process, we can describe the tasks conducted by ontologist leaders as individual ones. The process follows the main phases of most ontology development methodologies [Jones *et al*., 1998]. The guidelines for these tasks that need to be performed for each ontology are the following:

- Define the basic terms of the concepts pertaining to the ontology and provide definitions in natural language to remove ambiguity. Definitions of terms have to be agreed among participating nations and validated with domain experts (in our case, military people);
- Build a preliminary ontology: formally specify the semantics of the concepts by describing their properties and relationships with other concepts. This should be done using the agreed formalism (meta-model) chosen in the previous stage;
- Publish the ontology to get comments from other nations;
- Integrate comments from other nations;
- Validate the final model with other nations; and
- Encode the ontology in the chosen language.

The whole three-stage methodology to support collaborative development of ontologies is illustrated in Figure 3.

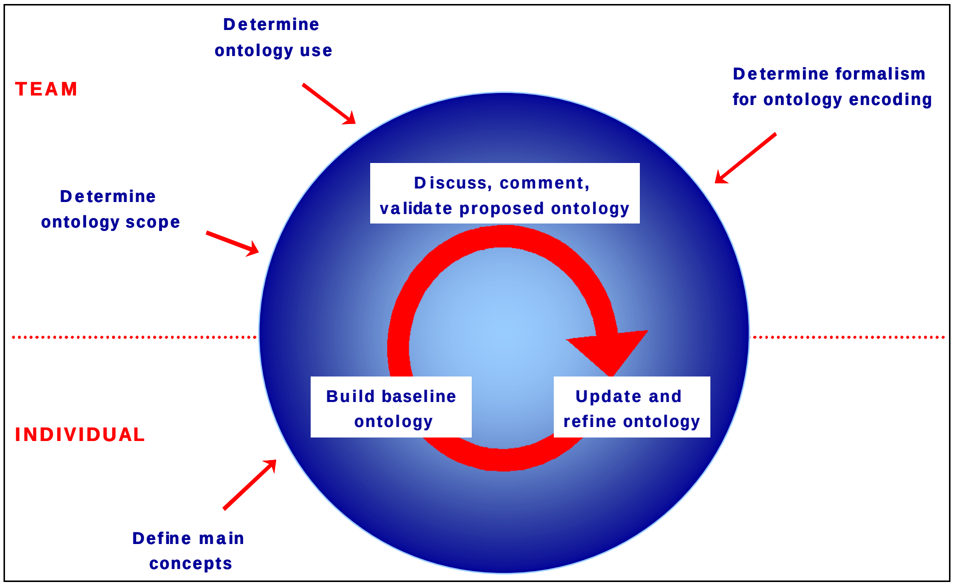

Figure 3. Methodology to collaborative ontology development

Our approach promotes ontology development at a conceptual level first, followed by the translation of the ontology into a specific language in order to satisfy ontology-based services (e.g. information exchange). In Figure 4, we illustrate the ontology conceptual levels on the left triangle, i.e. the ontology meta-model, the ontology model, and ontology instances from various information sources. The right triangle shows their instantiation using the selected formalism (e.g. XML-Schema is selected to encode the ontology in the context of Internet technology).

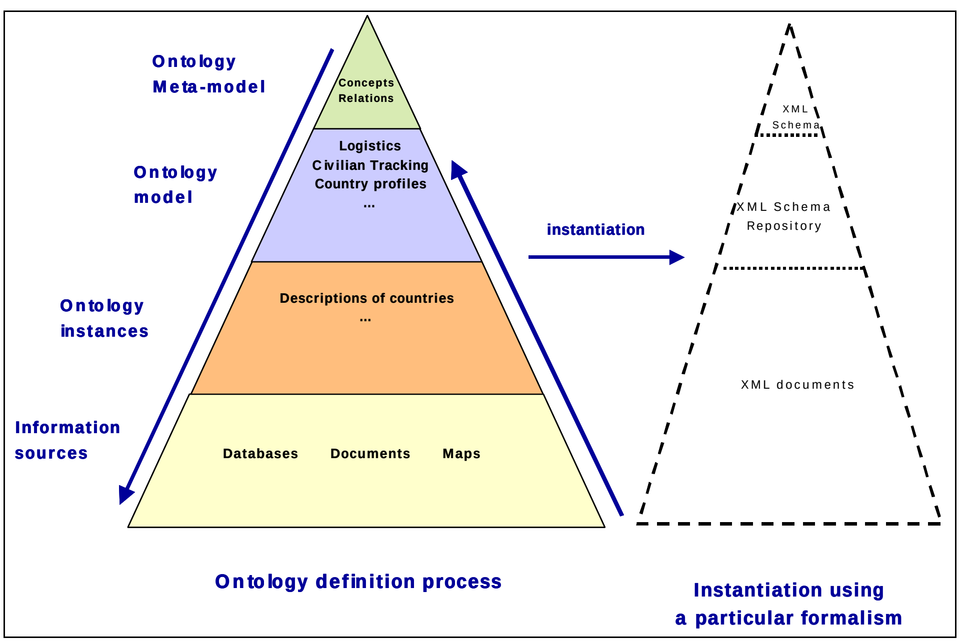

Figure 4. Conceptual levels and instantiation of an ontology

4. **Ontology engineering tools**

Ontology engineering tools provide functions for editing, browsing and visualizing ontologies through user-friendly interfaces. Several tools have been proposed from the Artificial Intelligence (AI) community for the last decade. New environments are now proposed in close relation to new ontology languages, in particular to develop the Semantic Web [Helfin *et al*., 2002].

Given the objective of collaboratively build and implement coalition ontologies, some requirements for an ontology development tool can be formulated as follows:

- The environment should help support the methodological process of building shared ontologies. In particular, it should allow an incremental development of ontologies.
- The environment should be user-friendly to facilitate the ontology development process by non-specialists, and validation by military people. It should facilitate both the building of ontologies and their exploitation (browsing, search for concepts, etc.).
- The environment should provide the appropriate knowledge representation formalism to represent ontologies. This one should depend on the expressiveness required, according to the role of the ontology. Consequently, the tool should provide some built-in primitives, or a flexible meta-model.
- The environment should provide functionalities related to collaborative work in order to facilitate the building of ontologies by different people in different locations. It should include the possibility to create groups of users with specific access rights, and provide a shared space to discuss the ontology under development, etc.
- The environment should provide import-export facilities in order to be able to reuse existing ontologies and to export the ontology being built in a specific formalism (e.g. XML or RDF Schema).
- The environment should also facilitate the building of a knowledge base relying on the ontology (e.g. country profiles and their instances).

Based on the results of a comparative study of ontology engineering tools [Duineveld *et al*., 99] and our knowledge and experiments with available tools, e.g. Protégé 2000 [Roy *et al*., 2000] or Ontolingua Server [Farquhar *et al*., 1996], we decided to design and implement our own tool in order to satisfy the requirements stated above.

5. **Description of OntoCINC Server**

The OntoCINC Server is based on the Teximus Expertise tool [Teximus]. Teximus Expertise stands in the category of web-based content management tools. It is a Web-based environment for the modeling of any domain content that facilitates the automatic generation of the Web pages presenting this domain based on the underlying knowledge model. In particular, one of the strengths of Teximus is that it allows users to define the meta-model needed for their application.

Consequently, the OntoCINC Server is a web-based collaborative environment that enables different people to develop a common ontology. It provides a flexible mechanism to freely specify a meta-model that will represent both an ontology description (concepts, attributes, relations) and collaborative aspects to facilitate discussion about the ontology under development (by adding issues, decision and related-questions properties to concepts).

The environment helps follow the proposed methodology to collaborative development of ontologies. It can be divided into three main parts as illustrated in Figure 5.

- Administration: to specify, update and manage the meta-model via appropriate views and to manage participants' access rights.
- Content authoring: to define the concepts, attributes (terms, synonyms, meanings), and relations to express the conceptual model of an ontology.
- Collaboration: to provide participants with means to discuss, comment and take decision during the ontology construction.

Figure 5. Ontology development environment

We present below the functionalities of the OntoCINC server according to the requirements stated previously, in particular knowledge representation and collaborative functions.

5.1 *Knowledge representation*

Usually, ontology development tools provide a predefined meta-model with built-in primitives (e.g. based on description logic or frame languages). The OntoCINC server provides the capability to define the ontology at a conceptual level, based on the meta-model defined by ontology builders. This is similar to the approach proposed within the ODE environment [Blasquez *et al.*, 1998]. Thus, the ontology definition is dependent on the meta-model defined, not on a particular formalism. This is easier to use for non-specialists and for validation by domain experts.

To meet our requirements, we created our own meta-model for ontology specification (using concepts, attributes, relations) and for ontology *critiquing* (by adding issues, decision and related-questions properties to concepts). The meta-model has been discussed with researchers of the C-CINC21 ontology group during a Workshop and is currently implemented within the OntoCINC server. Based on this meta-model, a concept in the ontology is defined by a class and has a definition, synonyms, and is characterized by a list of attributes. Predefined relations between classes are provided: *subsumes* (reverse relationship is kind-of), and *part-of* (reverse relationship is contains). In addition, any established relationship between classes can be named.

Moreover, each C-CINC21 nation representative can provide and specify within the model its own definition of a concept. But the reach of an agreement is necessary, since each concept must have a unique semantic meaning to ensure interoperability.

A graphical representation of the meta-model and a definition of a class within the OntoCINC server are respectively showed in Figures 6 and 7.

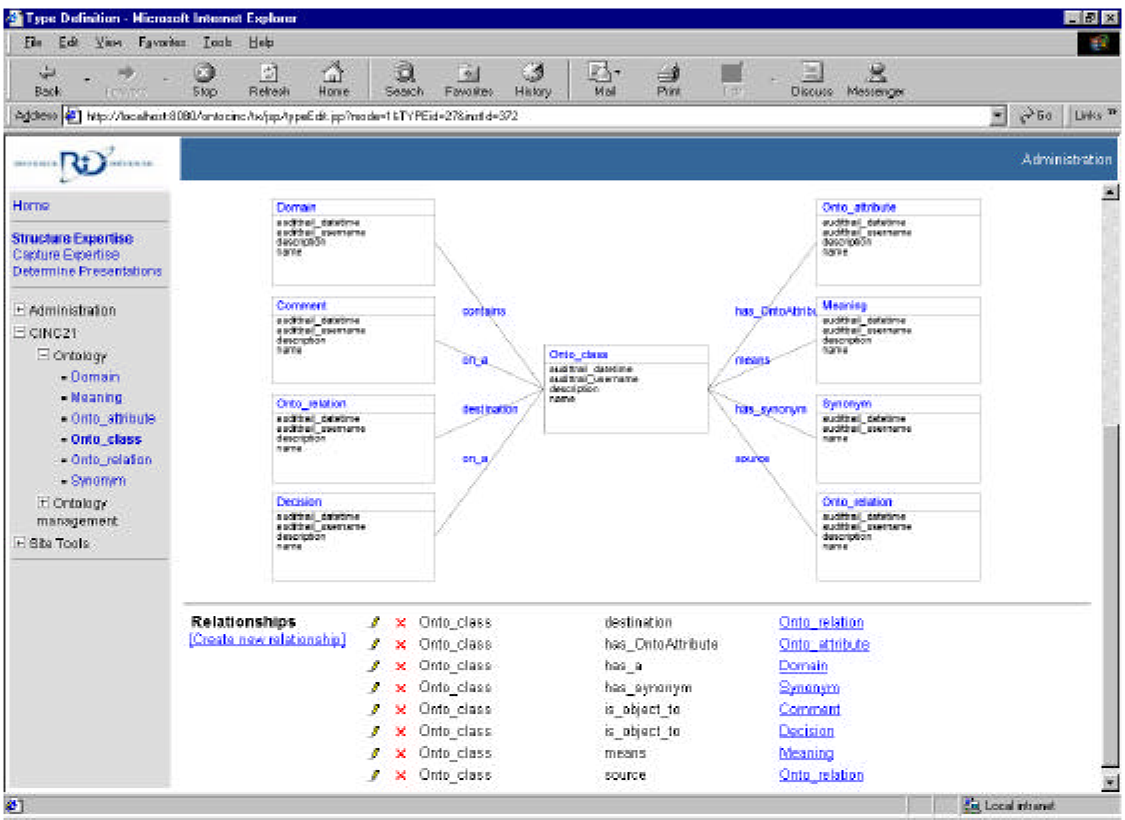

Figure 6. OntoCINC meta-model description

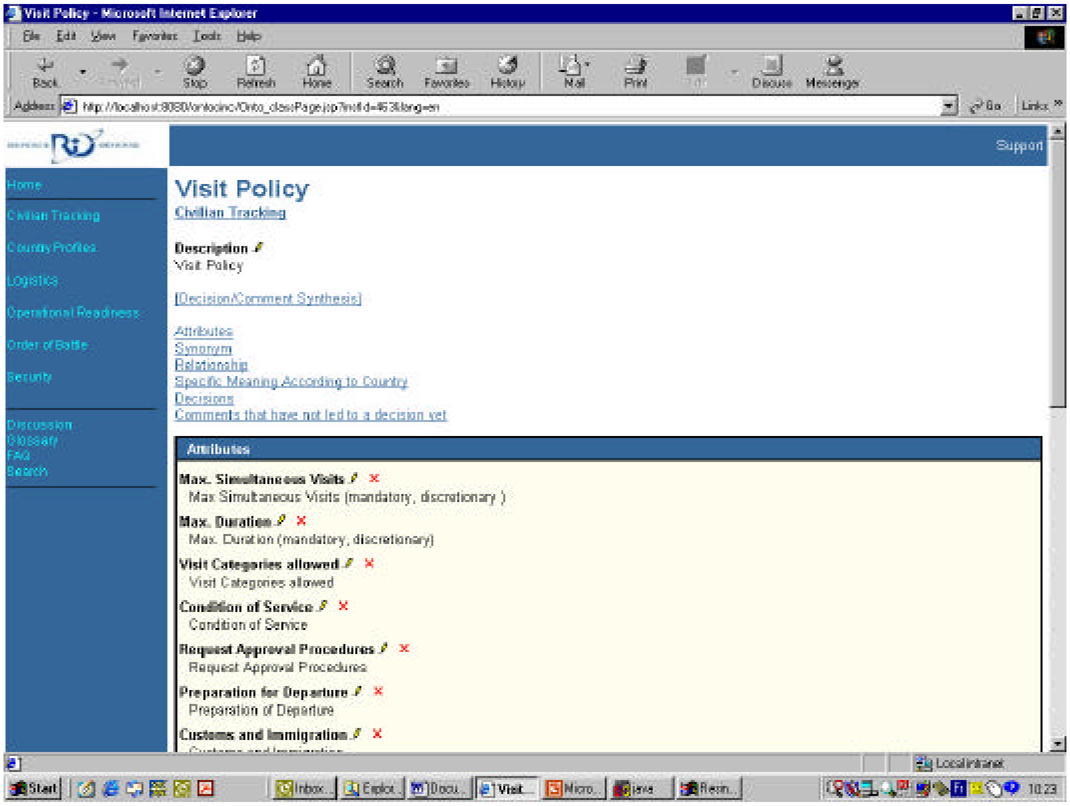

Figure 7. Class definition within OntoCINC

5.2 *Collaborative functions*

Among ontology engineering tools, some are standalone applications whereas others enable the collaborative construction of ontologies (e.g. Ontolingua Server, or WebOnto). Most of the tools provide synchronous cooperation by implementing lock functions and define groups of users with different Read-Write access rights. However, in a distributed environment, it is important for different ontologists to have a workplace to put their comments or discuss modeling issues about parts of ontologies that are built by other people. For example, one person shall question whether if a new concept (class) should be defined as a subclass of an existing one or as an instance of a class with its own properties. In another case, one has to decide whether if a proposed class should be divided into two distinct classes in order to better reflect the world being modeled. In this context, we have introduced the concept of *ontology critiquing* in order to enable discussions about modeling decisions. Therefore, in the perspective of providing a powerful collaborative ontology development tool, concepts such as *comments* or *decisions* related to the concepts of the ontology are integrated in the ontology meta-model.

Using such a configuration, each ontologist leader contributing to the ontology development process can build the ontology he/she is responsible for and get comments from other people efficiently. The tool automatically identifies the person who wrote a comment or made a change in the ontology and records it within the knowledge base. As other collaborative ontology development tools (e.g. Ontolingua server), the OntoCINC Server provides the possibility to assign different Read/Write access rights to users depending on their role in the ontology development process. Thus, one person can edit the ontology and perform changes whereas others could only browse it and add comments.

Figure 8 presents the user interface that contains the list of concepts and the comments and decisions related to them.

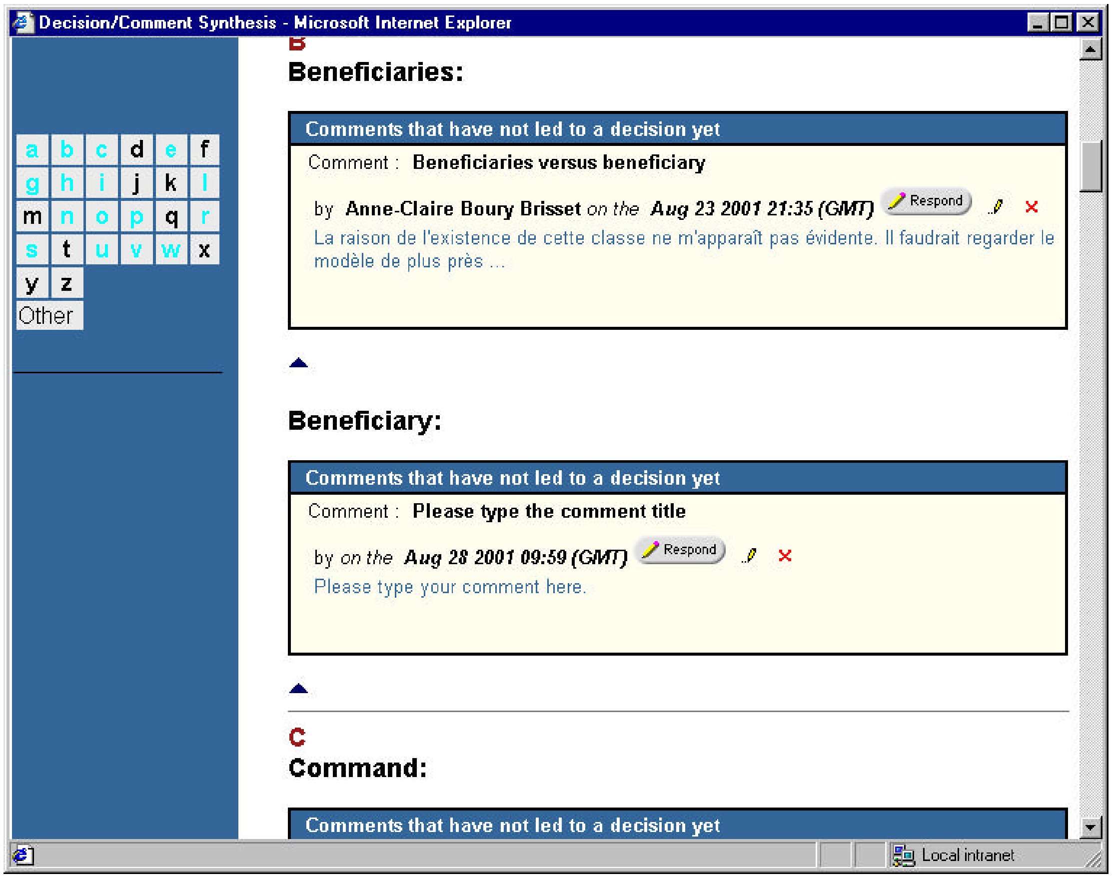

Figure 8. Comments during the development of the ontology model

5.3 *User interface*

The Teximus Expertise tool facilitates the generation of user-friendly interfaces. Given a defined meta-model, user interfaces can be provided, corresponding to the look and feel needed. Consequently, user interfaces can be easily customized to satisfy users preferences.

Graphical capabilities are provided to visualize the meta-model being built (Figure 6). From the user perspective, various interfaces are implemented to edit classes, relationships between concepts, or add comments. It is also possible to browse the ontology under development to visualize the concepts and look at the comments given by other people. Comments and decision about modeling issues can be accessed within the concept definition interface or in an interface providing a synthesized view of all comments/decisions.

5.4 *Export format*

Using the OntoCINC Server, once a meta-model has been agreed between users and a target encoding language identified (e.g. RDFS or DAML), a translation module should have to be provided to convert the ontology into the chosen encoding language. For example, Protégé 2000 records ontologies being built in text files but allows export formats, such as JDBC databases and RDF. For the last one, a module has been developed to translate Protégé knowledge models into RDF formalism. Whereas other concurrent products are oriented towards XML-based representation, the Teximus Expertise tool is knowledge-oriented, and XML is only considered as an export (or import) format.

The OntoCINC Server includes other useful functions. Among them, a search engine enables to rapidly search for information, for example, concepts along the class tree. Furthermore, a knowledge base of Frequently Asked Questions is provided and a forum discussion allows users to discuss general questions. The environment also supports the building and management of a knowledge base relying on the ontology model.

6. **Comparison with other tools**

In the previous section, we described the OntoCINC functionalities with regard to existing ontology development tools prompted within the AI community. In this section, we mention other categories of tools that have emerged more recently.

6.1 *XML-based editors*

When XML emerged a few years ago, many tools appear to deal with XML documents (parsers, editors, etc.), either from IBM, Microsoft or specialized companies. More recently, popular XML schemas editors are also proposed (e.g. XML Spy From Altova), or RDFedt, an RDF content editor, which includes a DAML Element Set plugin.

Even if there exist user-friendly tools in this category, these are designed to facilitate the building of XML or RDF schemas. Consequently, their knowledge representation expressiveness is limited to this formalism and they do not constitute an environment for collaboratively develop and discuss about ontologies.

6.2 *DAML*

From the DAML project (DARPA Agent Markup Language) [Hendler and McGuinness, 2000], different tools have emerged, such as a browser, crawler, transformation validator, viewer, inference engine, or ontology analyzer [DAML]. In particular, an ontology editor, DAML UML Enhanced Tool (DUET) provides a UML visualization and authoring environment for DAML. The initial implementation is an Addin to Rational Rose. OntoEdit is another environment recently proposed for collaborative ontology development for the semantic Web [Sure *et al*., 2002]. A new ontology language is currently being defined for the Web [Heflin *et al*., 2002] so that new formalism and related tools will emerge in the next future.

6.3 *Other categories of tools*

Collaborative tools such as Groove, Netmeeting could be used to develop shared ontologies in the sense that they provide a shared space for discussion where ontologists could “chat” about conceptual modeling. However, they are not well suited to the building of structured knowledge models, given that there is no knowledge representation capability underlying the software.

In some other cases, ontologies have been built using tools designed for other purposes. UML (Unified Modeling Language) is sometimes presented as a potential language for describing ontologies. Thus, case tools supporting UML (e.g. Rational Rose) could be considered as ontology building tools. But they do not offer the flexibility and capabilities required in our context as mentioned above.

7. **Lessons learned in applying the methodology within the coalition experiment initiative**

In the scope of the C-CINC21 initiative, we participated as members of the working group to collaboratively construct ontologies to facilitate coalition interoperability. The OntoCINC server has been implemented at DRDC - Valcartier and made available to all researchers of the group. Issues from the experience we gained in applying the methodology and using the OntoCINC server are described in regards to the stages of the overall collaborative ontology development process.

7.1 *Stage 1: Identification of domain areas and role attribution*

Because of the experimental objective of the C-CINC21 initiative, researchers identified the domain areas of military business ontologies from an insightful viewpoint driven by their experience and national leadership in certain areas rather than from a pure structuring of the “coalition universe” as military experts would have done. Hence, military business ontologies and lead nations were agreed as illustrated in figure 9.

Figure 9. Military business ontologies

The support ontology on security was assigned to Australia. An interesting source of information for building the country profiles ontology is the CIA World Fact Book [CIA WFB].

7.2 *Stage 2: Selection and agreement on a formalism, an engineering tool and a meta-model*

At the time to specify the meta-model in which all ontologies will be represented and specified, we encountered difficulties in agreeing on a final meta-model that fulfills the vision that each researcher has on this topic. Finally, to ease the task and to follow a step-by-step learning approach, we agreed on a simplified meta-model to represent subsequent ontologies that has only one layer (as a web of relations), where the hierarchy of concepts is expressed by the relations (e.g. subsumes, kind-of). The notion of subclass will not be instantiated in the specification of the ontologies. Hence, the meta-model that we (as the whole working group) specified with the tool will be inadequate to support the implementation of complex ontologies, although the tool provides the flexibility to define a more powerful meta-model. However, the choice of this simplified meta-model still permitted to evaluate the collaborative aspect in the construction of ontologies. It was decided to review the specification of the meta-model as more complex ontologies will be addressed.

7.3 *Stage 3: Development of ontologies*

The OntoCINC server has been experimented to build a limited version of the civilian-tracking ontology, focusing on the non-combatants particularly. At the time to specify the model, we learned interesting lessons on the complexity of collaborating in the construction of ontologies, even for a rather small ontology. We acknowledged the real challenge brought by the fact that participants have different background and culture and disparate knowledge of the military doctrine (either tacit or explicit). For instance, the concept of VISITOR was not obvious when we had to formally specify it. Its intrinsic meaning is different between nations and for others, the concept even does not exist as such, it is a state attribute of a more general concept.

Also, participants deplored the lack of appropriate graphical representation of the ontology model offered by the Teximus Expertise tool and had for effect to lessen their enthusiasm towards the use of the OntoCINC server.

8. **Conclusion**

We proposed in this paper a methodology to collaborative development of ontologies. We presented the OntoCINC Server that aims to satisfy the requirements of collaboratively design coalition ontologies. The proposed environment supports some collaborative functions, is user-friendly and flexible. In particular, several functionalities can be provided and adapted when configuring the development environment (e.g. ontology meta-model building, interface customized to users' needs). The tool was experimented to build a limited version of the civilian-tracking ontology. This is not sufficient to validate its ability to properly represent ontologies. In particular, it would be interesting to develop larger ontologies, translate them into an appropriate formalism in order to exploit information exchange services based on them

Apart from imposed security constraints within the Defence environments and slow response time across the network, we learned from both the ontology development process and the ontology engineering tool. Comments from users allow us to improve some of the functionalities and user interfaces. The methodology has proven to be appropriate.

9. References

[Blasquez *et al*., 1998] M. Blazquez, M. Fernandez, J. M. Garcia-Pinar, A. Gomez-Perez. Building ontologies at the knowledge level using the ontology design environment, In Proceedings of the 11th International Workshop on Knowledge Acquisition, Modeling and Mangement (KAW'98), Banff, Canada, October 1998, 1998.

[Boury-Brisset, 2001] A.-C. Boury-Brisset, Ontologies for coalition interoperability, in Proceedings of the NATO RTO Symposium (MP-064), Information Management Challenges in Achieving Coalition Interoperability, Québec, Canada, May 2001.

[DAML], The DARPA Agent Markup Language, http://www.daml.org.

[Duineveld *et al*., 1999] A. J. Duineveld, R. Stoter, M. R. Weiden, B. Kenepa, V. R. Benjamins, Wondertools? A comparative study of ontological engineering tools, in Proceedings of the Workshop on Knowledge Acquisition, Modeling, and Management, KAW'99, Banff, Canada, 1999.

[Farquhar *et al*., 1996] A. Farquhar, R.Fikes, and J. Rice, The Onlingua Server: A Tool for Collaborative Ontology Construction, in Proceedings of Knowledge Acquisition Workshop, Banff, Canada, 1996.

[Gruber, 1993] T. Gruber, A translation Approach to Portable Ontology Specifications, in Knowledge Acquisition, 5(2), pp. 199-220, 1993.

[Heflin *et al*., 2002] J. Heflin, R. Volz, J. Dale, Requirements for a Web Ontology Language, Working draft of the W3C Ontology Working Group, http://www.w3c.org/TR/webont-req.

[Hendler and McGuinness, 2000] J. Hendler, D. McGuinness, The DARPA Agent Markup Language, in IEEE Intelligent Systems, Vol. 15, No. 6, Nov/Dec. 2000.

[Holsapple and Joshi, 2002] C. W. Holsapple, K. D. Joshi, A Collaborative Approach to Ontology Design, in Communications of the ACM, Vol. 45, No. 2, February 2002.

[Jones *et al*., 1998] D. Jones, T. Bench-Capon, P. Visser, Methodologies for Ontology Development, in Proceedings of IT&KNOWS (Information Technology and Knowledge Systems) of the 15th IFIP World Computer Congress, Budapest, Hungary, 1998.

[McGuinness, 2000] D. L. McGuinness, Conceptual Modeling for Distributed Ontology Environments, In Proceedings of the Eighth International Conference on Conceptual Structures Logical, Linguistic, and Computational Issues (ICCS 2000), Darmstadt, Germany. August 14-18, 2000.

[Roy *et al*., 2000] N.F. Roy, R. W. Fergerson, M. A. Musen. The knowledge model of Protege-2000: Combining interoperability and flexibility, in Proceedings of the 2nd International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000), Juan-les-Pins, France, 2000.

[Sure *et al*., 2002] Y. Sure, M. Erdmann, J. Angele, S. Staff, R. Studer, D. Wenke, OntoEdit: Collaborative Ontology Development for the Semantic Web, in Proceedings of the International Semantic Web Conference (ISWC) 2002, Sardinia, Italia , June 9-12 2002.

[Teximus] Teximus Expertise Fact Sheet, http://www.Teximus.com.