Bart-Jan van Putten, Shawn Wolfe, Virginia Dignum: An Ontology for Traffic Flow Management. ATIO'08, Anchorage, Alaska, September 2008

An Ontology for Traffic Flow Management

Bart-Jan van Putten
USRA RIACS /
Utrecht University
The Netherlands
bputten@cs.uu.nl

Shawn R. Wolfe
Intelligent Systems Division
NASA Ames Research Center
Moffet Field, CA, U.S.A.
Shawn.R.Wolfe@nasa.gov

Virginia Dignum
Dept. of Information and
Computing Sciences
Utrecht University
The Netherlands
virginia@cs.uu.nl

1. Introduction

The Next Generation Air Transportation System (NGATS) project is a multi-faceted research effort to address issues with the National Airspace System. One such facet is the area of Collaborative Traffic Flow Management (CTFM), which intends to increase both the efficiency of the National Airspace System and the satisfaction level of the airlines. In today's system, the flow of traffic is primarily handled by three entities: the FAA's Air Traffic Control System Command Center (ATCSCC), Traffic Management Units (TMUs), and the individual airlines' Airline Operation Centers (AOCs). Previous field observations found that several aspects of the current system hindered collaboration [1], leading to the development of a new concept of operations for CTFM [2].

In order to develop a new concept of operations, a better understanding is needed on its consequences for the work processes, interdependencies and capabilities of the parties involved. We will use an agent-based social simulation approach to evaluate the concept of operations [3]. Agents in the simulation are autonomous pieces of software that represent the proposed work practice of participants in TFM. Agents are able to make their own decisions based on their individual understanding of the situation. The resulting overall behavior of system is achieved through the interaction of the different agents.

Communication is one of the core aspects in agent systems, that is, it is crucial for the agents to have mutual understanding about how meanings are expressed in messages. Ontologies enable this shared understanding. Agents that have adopted the same ontology have no difficulty in sharing knowledge as the common ontology guarantees that the receiver's interpretation of a message corresponds to what the sender intended to convey with it [4]. In order to formally support their conceptualization of the domain, that enables communication between agents, we are developing an ontology for Traffic Flow Management. In this paper, we describe this ontology in more detail.

The paper is organized as follows. In section 2 we provide an overview of the research field and its applicability to the TFM domain. Section 3 describes relevant existing ontologies. In section 4 we describe the ontology that we have developed for this purpose and present results of its evaluation. Finally, in section 5, we indicate directions for future development and we present our conclusions in section 6.

2. Ontologies

To be able to understand a domain, such as TFM, people need a conceptualization of that domain. The result of a conceptualization is a set of concepts (i.e. classes), properties of those concepts (i.e. metadata) and relationships between those concepts. According to Gruber [5], a conceptualization is "an abstract simplified view of the world that we represent for some purpose". An ontology is a way of expressing that conceptualization. According to Borst and Akkermans, an ontology is "a formal, explicit specification of a shared conceptualization" [6]. *Formal* means that the ontology should be machine-readable. *Explicit* means that the type of concepts used, and the constraints on their use are explicitly defined. *Shared* means that an ontology is often meant to capture knowledge that is accepted by a group.

Ontologies bridge the gap between systems used by people to understand a domain, and systems used by computers to reason about (i.e. 'understand') a domain. Ontologies can be used by people because they can describe the rich variety of human cognition in terms of concepts, properties, relations, assertions, and reasoning. Ontologies can also be used by computers, because they are machine interpretable, enabling them to understand the semantics and derive new knowledge. An example: this apple (concept) is red (value of property 'color'), an apple is a fruit (relation), I like sweet fruits (assertion), red apples are sweet (assertion), so I like this apple (reasoning).

Ontologies are a type of semantic structures. Other possible ways of representing a domain are controlled vocabularies, thesauri, and taxonomies [7]. However, such representations are mostly geared to human consumption and are less appropriate to support machine-based reasoning. In the following, we briefly describe the characteristics and applicability of the different structures.

Controlled Vocabulary

A controlled vocabulary is a list of allowable words (concepts) for a certain context in which they may be used. However, the vocabulary does not express any relationships between the concepts, unlike most other semantic structures, including ontologies.

Thesaurus

A thesaurus links similar concepts by a predefined set of relationships, such as synonymy. Though useful in many contexts, the scope of encoded knowledge in a thesaurus is limited by the types of relationships that it can express. This is in contrast to ontologies, which can express any number of relationships at the same time.

Taxonomy

Taxonomies organize concepts in a hierarchical structure, often for the purpose of classification. Taxonomies are limited to defining only one relationship between elements, typically either inheritance or composition. An example of the inheritance relationship: an apple *is a* fruit, an orange *is a* fruit. An example of the composition relationship: a car *consists of* wheels, chassis.

Ontologies are semantically richer than controlled vocabularies, thesauri, and taxonomies because they can be used to express any number of different relationships between concepts. Therefore, they are more useful for expressing rich semantics. Additionally, because ontologies can express this richness in a formal way, they are better suited for making knowledge machine-readable.

3. Related Ontologies

Ontologies are used for many purposes where people, computers, or both, need conceptualizations of a domain. One of the purposes stated by Noy and McGuiness is "sharing common understanding of the structure of information among people or software agents" [8]. Reiss et al. also address the use of ontologies for building of scenarios for simulation and training [9]. Several ontologies have been developed and used in other space and aviation applications:

- CRIISTAL: Several ontologies were developed for indexing and search of documents in the aeronautics domain. One application of the ontology was to link documentation to scenarios for working with the Traffic alert and Collision Avoidance System. [9]
- AIAA Topic Database: This is a controlled vocabulary in the aviation domain, developed to support shared understanding on topics that are of interest to the community. [10]
- XCALIBR: This is a spacecraft ontology, used for the 'plug and play' assembly of spacecraft. This is possible because the ontology describes relations between components, supporting spacecraft designers in determining the dependency of components on other components. [11]
- MillInfo: This ontology describes military information, to be used as a foundation for semantic architecture models for airspace systems. [12]
- TSONT: This is an ontology for (military) trajectory simulation. The aim of the general research is the construction of a re-use infrastructure to be used in the development of a variety of trajectory simulations. The ontology is regarded as the domain model component of the reuse infrastructure. The reuse of knowledge saves time and money when building new simulations. [13]

4. An Ontology for Traffic Flow Management

Unfortunately the existing ontologies do not cover TFM. Therefore a new ontology was developed, to serve our project in the following ways:

1. *Knowledge Acquisition:* The goal of our project is to simulate the concept of operations. Before we can simulate, we need to model the current and projected TFM behavior. In order to do that, we need to acquire and store domain knowledge. The TFM ontology is used to record this domain knowledge in terms of concepts, relations, and assertions. This way, we can store knowledge about the domain before it is used further by people or systems.
2. *Software Engineering:* The TFM ontology is primarily used as the knowledge structure of our OperA and Brahms agent-based models [14, 15]. These agent-based models require rich semantic structures to be able to reason with the data.

The ontology is capable of expressing many different kinds of concepts, relationships, and assertions, and is therefore very suitable for this goal.

3. *Training:* The domain of TFM is complex: there are many different organizations involved, there are many procedures, specialist terms, acronyms, and expert tasks. This makes the domain difficult to comprehend, which is a problem for new employees who start to work in this area. By allowing those people to take part in the development of the TFM ontology, they are actively constructing mental models of the domain, which facilitates comprehension, retention, and problem-solving capabilities [16].

From our perspective, the TFM ontology has several benefits for the TFM community:

1. *Support communication:* The TFM community involves many different parties, the ATCSCC, TMUs, AOCs, pilots, and more. All these parties have (slightly) different goals, procedures, and as such, a different perspective on the domain. This makes communication among those parties more difficult, while more collaboration is one of the current goals of TFM. The TFM ontology can support different parties to share their understanding of the domain, and to make it more explicit, which enables better communication and collaboration.
2. *Support information systems:* More and more information that is being used for TFM has been digitized; also more and more processes are being automated. For successful collaboration in TFM the information systems need to be interoperable. If the different parties can agree on a shared conceptualization of the domain, the TFM ontology can be used to represent this conceptualization and function as the basic data structure underlying to the information systems.

Design

The TFM ontology was designed using Protégé [17], the de-facto standard in ontology engineering tools. This tool allows you to specify classes (such as ‘fruit’), and sub-classes (such as ‘apple’). Sub-classes inherit the properties of super-classes, and can be used to describe things in more detail. The tool also allows you to define relationships between classes, e.g. fruits *have a* taste. Protégé can store the ontology in different formal languages, which influences the level of semantic richness of the things that you can express. OWL [18,19,20] is currently the most advanced ontology language, and was therefore selected to build our ontology. OWL for example allows the specification of assertions about classes and relationships, e.g. ‘red apples taste sweet’ (IF apple.color = red THEN apple.taste = sweet). In this section we will show some examples and some main components of the TFM ontology.

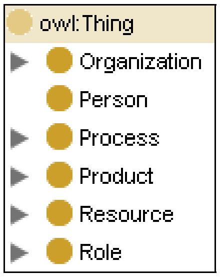
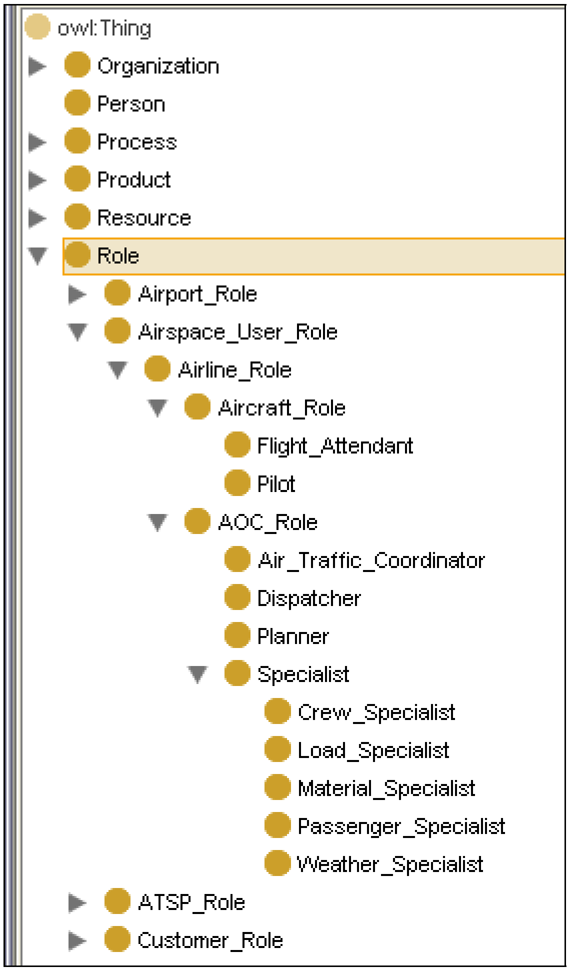


Figure 1: Upper-level

Figure 1 shows the upper-level of the ontology. It consists of six concepts that relate to each other in the following way: "*people* are members of an *organization*, they enact *roles* and participate in *processes*, using *resources*, and producing *products*". This upper-level was used to convey the lower levels in a comprehensive manner. They abstract from the vast amount of subclasses (e.g. figure 2), to make the structure more user-friendly for people that want to use or develop the ontology.



- owl:Thing
  - ▶ Organization
  - Person
  - ▶ Process
  - ▶ Product
  - ▶ Resource
  - ▼ Role
    - ▶ Airport_Role
    - ▼ Airspace_User_Role
      - ▼ Airline_Role
        - ▼ Aircraft_Role
          - Flight_Attendant
          - Pilot
        - ▼ AOC_Role
          - Air_Traffic_Coordinator
          - Dispatcher
          - Planner
          - ▼ Specialist
            - Crew_Specialist
            - Load_Specialist
            - Material_Specialist
            - Passenger_Specialist
            - Weather_Specialist
    - ▶ ATSP_Role
    - ▶ Customer_Role

Figure 2: Lower-level: Role

The upper-level of the TFM ontology consists of the following classes:

- *Organization:* Describes the organizations that are involved in TFM. We distinguish between *air traffic service providers* (ATSPs) and *airspace users*.
- *Person:* This class has no subclasses, it can just be instantiated. Most properties of persons depend on the role that they enact and are therefore covered by the top-level: *role*.
- *Process:* Describes the processes involved in TFM, divided over the four main phases: *constraint identification*, *impact assessment*, *flow planning*, and *flight implementation*.
- *Product:* Describes the products that result from TFM processes. Examples of such products are *flight plans*, and *weather status reports*.

- *Resource:* Describes the resources used in the TFM processes. Examples of such resources are *air facilities*, *vehicles*, and *information systems*.
- *Role:* Describes the roles involved in the execution of the TFM processes. Examples of roles are *pilot*, *air traffic coordinator*, and *weather specialist*.

Methodology

The design of the TFM ontology is part of our simulation design methodology, which consists of the following steps:

1. *Knowledge Acquisition:* Field observations, expert knowledge, and related work were used to gain understanding of the domain.
2. *Knowledge Representation:* The domain knowledge was expressed in the ontology in terms of concepts, relations, and assertions. For this step we used a methodology similar to the ones described in [8,21].
3. *Use in OperA and Brahms:* The ontology is integrated with the OperA and Brahms agent-based models [14,15].
4. *Verification and Validation:* The models just described are verified, validated, and improved in an iterative way. In the end, the Brahms environment allows us to run simulations and validate the concept of operations.

Evaluation

The TFM ontology has been evaluated and improved on several dimensions:

- *Verification:* An ontology verifies [22] if the syntax (language) has been used correctly, when the acquired knowledge is represented by the right constructs in the ontology, and when the parts of the ontology are consistent: an assertion can only assert things about concepts that actually exist in the ontology.
- *Validation:* An ontology is valid [22] if the semantics (i.e., meaning) are expressed correctly, e.g. if the ontology states that an aircraft is part of the airline organization, the semantics are not correct, as the relationship between aircraft and airline should be expressed with an ownership relation rather than a composition relationship.
- *Usefulness:* A system is useful if it has value for the situation in which it is intended to be used [23]. Ontologies should neither consist of too many concepts nor of too few concepts. On one hand an ontology with too few concepts is not capable to support the applications it was built for. An ontology with too many concepts however makes it difficult to maintain the quality, for example in terms of consistency of the structure, and novelty of the information. Therefore the usefulness of an ontology is determined largely by the amount of concepts.
- *Usability:* A system is usable if it is easy to operate [23]. A lot of usability research has been done in the area of hypertext systems, like web sites. A rule of thumb is that there should be no more than six concepts on one level of a hierarchical information structure within one branch of the tree. We believe that this rule also applies to the design of the structure of ontologies, to make sure that they are comprehensible and usable by people.

Lessons Learned

Currently, the development of the TFM ontology is in an early stage. It has been *verified*, and it has been *validated* with a subject matter expert. When the agent-based simulation is finished it will also be possible to evaluate the *usefulness* of the ontology. The *usability* of the TFM ontology is continuously monitored and can be evaluated when new employees start working in the area of TFM. From our *usability* evaluations we have learned some lessons, which are partly similar to the lessons learned by some other applications of ontology engineering [24,10,12]:

- *Add descriptions:* For people who did not design the ontology, the labels of the classes, properties, and relationships are often not sufficient to understand it. Descriptions should therefore be added wherever they are necessary.
- *Limit number of concepts on one level:* The structure of an ontology is easier to comprehend if the number of concepts on one level within a branch of the class hierarchy is limited.
- *Limit granularity:* If there are many classes which are similar, it is more difficult for people to understand the difference between those classes. The level of detail should therefore be limited to the minimum that is needed.
- *Add a thesaurus:* Although the different parties that use the ontology might need to agree on the semantics of a concept, they do not necessarily need to use the same term to refer to that concept. For example, the concept of an organization who makes use of the airspace, can be referred to as an ‘airspace user’, an ‘air traffic user’ or a ‘user’. All those terms are alright, as long as they point to the same concept. This way, organizations can exchange knowledge with the same semantics, but still use their preferred terminology.

5. Future Work

In the near future we want to develop the TFM ontology further. This can be done in at least the following three ways: (1) currently the ontology only contains enough concepts for one common problem scenario in TFM; the ontology may be extended in such a way that it incorporates multiple scenarios, (2) the collaborative design and evolution of the ontology can be supported [25], and (3) the ontology can be made available to other software engineering projects at NASA, and to the TFM community.

6. Conclusion

We have developed an ontology for Air Traffic Flow Management. Its main purpose is to enable reasoning and communication among software agents in a simulation of the concept of operations for Collaborative Traffic Flow Management. Additionally, the ontology has supported people in capturing and understanding the domain. In this paper we have presented the initial design and evaluation of this ontology; it has been verified and validated. Its usefulness can be determined when our simulation is finished. With respect to its usability, we have presented some lessons learned.

7. References

[1] Idris, H., et al. (2006) Field Observations of Interactions Between Traffic Flow Management and Airline Operations. In AIAA 6th Aviation, Technology, Integration, and Operations Conference (ATIO’06). Wichita, Kansas.

[2] Idris, H., et al. (2005) Operational Concept for Collaborative Traffic Flow Management based on Field Observations. In AIAA 5th Aviation, Technology, Integration, and Operations Conference (ATIO’05). Arlington, Virginia.

[3] Wolfe, S. R. (2007). Supporting Air Traffic Flow Management with Agents. Paper presented at the AAAI Spring Symposium: Interaction Challenges for Intelligent Assistants, Stanford, CA.

[4] Dignum, V., & Van Diggelen, J. (Eds.). (2006). Special Issue on Agent Mediated Knowledge Management (Vol. 10 (4)): IOS Press.

[5] Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 5(2).

[6] Borst, W. N., & Akkermans, J. M. (1997). Engineering Ontologies. International Journal on Human-Computer Studies, 46(2/3), 365-406.

[7] Schwarz, K., Kouwenhoven, T., Dignum, V., & Van Ossenbruggen, J. (2005). Supporting the decision process for the choice of a domain modeling scheme. Paper presented at the 1st workshop of Formal Ontologies Meet Industry (FOMI’05).

[8] Noy, N. F., & McGuinness, D. L. (2001). Ontology Development 101: A Guide to Creating Your First Ontology (Technical Report No. SMI-2001-0880).

[9] Reiss, M., Moal, M., Barnard, Y., Ramu, J.-P., & Froger, A. (2006). Using Ontologies to Conceptualize the Aeronautical Domain. Paper presented at the International Conference on Human-Computer Interaction in Aeronautics.

[10] Marshall, J. R., & Morris, A. T. (2007). Organization’s Orderly Interest Exploration: Inception, Development and Insights of AIAA’s Topics Database. Paper presented at the AIAA Infotech@Aerospace.

[11] Neff, J. M., Some, R., & Lyke, J. (2007). Lessons Learned in Building a Spacecraft XML Taxonomy and Ontology. Paper presented at the AIAA Infotech@Aerospace.

[12] Valente, A., Holmes, D., & Alvidrez, F. C. (2005). Using a Military Information Ontology to Build Semantic Architecture Models for Airspace Systems. Paper presented at the IEEE Aerospace Conference.

[13] Durak, U., Oguztuzun, H., & Ider, S. K. (2006). An ontology for trajectory simulation. Paper presented at the 38th Winter Simulation Conference.

[14] Dignum, V., Dignum, F., & Meyer, J. J. (2004). An Agent-Mediated Approach to the Support of Knowledge Sharing in Organizations. Knowledge Engineering Review, 19(2), 147-174.

[15] Sierhuis, M., Clancey, W. J., & Van Hoof, R. (2007). Brahms: A multiagent modeling environment for simulating work processes and practices. International Journal of Simulation and Process Modelling, Inderscience Publishers, 3(3), 134-152.

[16] Mayer, R. (1991). Multimedia aids to problem-solving transfer. International Journal of Educational Research, 31, 611-623.

[17] Protégé. (2007). http://protege.stanford.edu

[18] Antoniou, G., & Van Harmelen, F. (2004). Web Ontology Language: OWL. In Handbook on Ontologies: Springer.

[19] Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. Scientific American.

[20] Web Ontology Language (OWL). (2007). http://www.w3c.org/TR/owl-features

[21] Prieto-Diaz, R. (2003). A Faceted Approach to Building Ontologies. In Proceedings of the International Conference on Information Reuse and Integration (IRI‘03), Las Vegas, NV.

[22] Vermesan, A. I. (1998). Software Certification for Industry - Verification and Validation Issues in Expert Systems. Paper presented at the 9th International Workshop on Database and Expert Systems Applications (DEXA'98).

[23] Landauer, T. (1995). The Trouble with Computers: usefulness, usability, and productivity. Cambridge: The MIT Press.

[24] Carvalho, R.M., Berrios, D.C., Wolfe, S.R., & Williams, J. (2005). Ontology Development and Evolution in the Accident Investigation Domain, IEEE Aerospace Conference, Big Sky, MT.

[25] Bao, J., Caragea, D., & Honavar, V. (2006). Towards Collaborative Environments for Ontology Construction and Sharing. Paper presented at the International Symposium on Collaborative Technologies and Systems.