Ontology Development for Intelligent Information Logistics in Transportation

Hasan Koç, Birger Lantow, and Kurt Sandkuhl

The University of Rostock, Institute of Computer Science, Chair Business Information Systems Albert-Einstein-Str. 22, 18057 Rostock, Germany [hasan.koc, birger.lantow, kurt.sandkuhl] @uni-rostock.de

Abstract. Technological innovations in the area of wireless sensor networks, which allow for features like spontaneous networking and self-organization, are enablers for new kinds of IT services in many application domains. In order to fully exploit the potential of these technologies various industries show examples for innovations on the level of service management as well as with respect to the underlying business models. Based on a case study from transportation, this paper shows how ontologies can be used as the basis for new types of IT services. The focus during ontology development in this context is on creating an adaptable knowledge base for different kinds of services and to prepare for self-organization of the overall solution. The contributions of this paper are (a) an ontology for the field of information logistics services in transportation, (b) experiences from the development process based on a real-world scenario and, (c) potentials and limits of the ontology to accommodate features required for self-organization.

Keywords: Information Logistics, Ontology Engineering, Transportation Service, Self-Organization, Situation Awareness.

1 Introduction

During the last years, technological innovations in the area of wireless sensor networks have established themselves as enablers for new kinds of IT services in many application domains. In order to fully exploit the potential of these technologies, which offer features such as self-organization and spontaneous networking, various industries show examples for innovations on the level of service management as well as with respect to new kinds of products. Examples can be found in the area of functional products, wind turbines or factory automation. This paper investigates new kinds of services and the required knowledge base for an example of intelligent information logistics services in transportation and logistics. Information logistics aims at improving information flow in organizations by means of information systems.

The logistics industry has changed under the impact of the internal European market and of an increasing globalization into a high-technology industry, making intensive use of modern information technology. At the same time, the industrial demand for more dynamic logistics solutions with adequate IT support is increasing. Many industries experienced a shift in sourcing and logistics strategies from long-term

customer-supplier relationships to more networked strategies adapted for global markets, like value networks, flexible supply networks, cluster-based approaches up to on-demand cloud constellations.

Within the logistics industry, the transportation area is considered as promising application field for new types of intelligent information logistics services, since

- Advances in wireless sensor networks and sensor/actuator technologies allow for new ways of tagging and tracking goods and vehicles,
- Many different actors with heterogeneous information systems offer possibilities for automating or transforming processes by means of system integration,
- Due to growing requirements from environmental or security regulations, and an increasing awareness of sustainability issues on the customer side, the market for applications creating more ecological and economic services is developing fast.

Based on a case study from transportation, this paper shows how ontologies can be used as the basis for new types of IT services. The focus during ontology development in this context is on creating an adaptable knowledge base for different kinds of services and to prepare for self-organization of the overall solution. The contributions of this paper are (a) an ontology for the field of information logistics services in transportation, (b) experiences from the development process based on a real-world scenario and, (c) potentials and limits of the ontology to accommodate features required for self-organization.

The remaining part of the paper is structured as follows: section 2 summarizes the background for the work from the areas of ontology engineering and information logistics. Section 3 introduces the industrial case study including requirements to the knowledge base. Section 4 describes the ontology engineering process performed and presents the actual ontology. Section 5 investigates potentials and limits of the ontology regarding self-organization. Section 6 summarizes the work and draws conclusions.

2 Background

As a background for the work presented in this paper, we will describe relevant work in the areas of ontology engineering, information logistics and self-organization.

2.1 Ontology Engineering

Ontologies became popular in the 90's mostly in the Knowledge Engineering Community. There have been several definitions for what an ontology is. For the purposes of this article [2] provides the most suited definition "An ontology is a formal explicit specification of a shared conceptualization."

There has been a series of approaches proposed for developing ontologies. Despite the fact that the methodologies for ontology development have been subject to research during a number of years¹, there is no one 'correct' way or methodology for developing

¹ Detailed information about the ontology development methodologies can be found in [4,5]

ontologies [3, 4] Noy and McGuinness proposed in [4] an iterative ontology development process consisting of seven steps. In this work the Ontology for Trailer Surveillance (OTS) is being developed following their methodology as well as extending it by two more steps (create rules, create defined classes).

The approach of Noy and McGuinness consists of the following steps:

- Determine the domain and scope of the ontology: This is the starting point of ontology development. Several questions should be answered, i.e. "What is the domain that the ontology will cover?" or "for what we are going to use the ontology?" These questions should be populated and formed more specifically regarding the domain of interest in order to put together a list of "competency questions".
- Consider reusing existing ontologies: For a particular domain and task it should be investigated, whether the existing ontologies could be reused and if yes, how.
- Enumerate important terms in the ontology: A list of important terms should be written down.
- Define the classes and the class hierarchy: These terms should be organized as classes into a hierarchical taxonomy. A top-down, bottom-up or a combination approach could be used for that purpose.
- Define the properties of classes: The internal structure of concepts should be specified.
- Define the facets of the slots: Based on the OWL language model this step corresponds with the specification of object properties and their characteristics.
- Create instances: The last step is creating individual instances of classes in the hierarchy and adding object property assertions.

This approach is extended applying two more steps. After creating instances, the *rules* for more powerful reasoning need to be formulated, which also provide a consistent knowledge base. Next, the concept of *defined classes* is applied, i.e. if an individual fulfils the necessary and sufficient conditions given by the defined class, then it is inferred to be a member of this class.

2.2 Information Logistics

The research field information logistics was established in the late 1990s and defined in [14]. The main objective is optimized information provision and information flow, based on information content, time of delivery, location, presentation and quality. The information logistics field focuses on improving the information flow by applying logistic principles to information supply. During the last decade, many IT applications have been developed implementing the objective of information logistics. Some of the applications are services providing bad weather warnings, traffic information or personalized news, and solutions for businesses in different domains like WIND service (weather information on demand), Smart-Wear (location-based information supply for mobile users) [14]. An essential concept in information logistics is the "information demand" which is defined by [10] as "...the constantly changing need for

current, accurate, reliable, and integrated information to support (business) activities, whenever and where ever it is needed."

This definition implies a number of aspects that must be considered while analysing information demand and when constructing information logistics services. Information demand should change as the task, roles and responsibilities, to which information demand is connected, change. The information should be relevant, current, accurate and reliable; otherwise it will contribute to information overflow. The information demand should be integrated with the business activities, as it is necessary to have a solid knowledge about the context in order to be aware of any changes of information demand that might happen. Whenever and where ever emphasize the importance of time and location while analysing the information demand [10]. A specific method for information demand analysis was developed and evaluated in a number of industrial projects [11].

2.3 Self-Organizing Systems

"A self-organizing system consist of a set of entities that obtains an emerging global system behaviour via local interaction without centralized control." [7] Besides emergence and decentralization, autonomy, adaptivity, self-maintenance, and optimization are common features of self-organizing systems [16].

Furthermore, self-organising systems are characterised by their capacity to spontaneously produce a new organisation in case of environmental changes [18]. These systems are particularly robust, because they adapt to changes, and are able to ensure their own survivability [18].

Research efforts in this area include: The EC FP6 Ambient Networks project offered a complete, coherent wireless network solution based on dynamic composition of networks. It provides access to any network through instant inter-network agreements. The EC FP7 project SENSEI aimed at integrating the physical with the digital world of the network of the future. It produced: (i) a scalable architectural framework; (ii) an open service interface and corresponding semantic specification; (iii) network island solutions consisting of a set of cross-optimised and energy aware protocol stacks; (iv) pan European test platform enabling large scale experimental evaluation of the SENSEI results. Goal of EC FP7 project SOCRATES (Self-optimisation and self-configuration in wireless networks) was the automation of wireless access network planning and optimization by the application of self-organisation methods.

The general components of a self-organizing system are (adapted from [18]):

- The *environment* in which the autonomous, individual entities (the agents) evolve
- *Agents*, which might be among others software agents, robots or sensor nodes
- Self-organisation *mechanisms* (rules) that describe the behaviour of the agents for organization management and task-fulfilment
- *Artifacts* that contain information provided by *agents* and *environment*. They can be used as a means of communication for management and task fulfilment purposes.

Negotiation models are key *mechanisms* of self-organising networks. The following general negotiation models are examples [6]:

- Different forms of spontaneous *self-aggregation*, to enable multiple distributed *agents* to collectively and adaptively provide a distributed service, e.g. a holonic (self-similar) aggregation.
- *Self-management* as a way to enforce control in the ecology of *agents* if needed (e.g. assignment of "manager rights" to an *agent*.
- *Situation awareness* organization of situational information and their access by *agents*, promoting more informed adaptation choices by them and advanced forms of stigmergic (indirect) interactions.

One of the early activities in this field was the DARPA project Self-Organizing Sensor Networks which addressed networks of self-aware, self-reconfigurable and autonomous sensor nodes. This project implemented a number of functionality which can be used as guidelines for what *mechanisms* have to be implemented for selforganization: The nodes involved in a self-organizing systems have to be capable to

- spontaneously create an impromptu network,
- assemble the network themselves,
- dynamically adapt to device failure and degradation,
- manage movement of sensor nodes/agents, and
- react to changes in task and network requirements.

The implementation of these capabilities can be realized by negotiation models like *self-aggregation*, *self-management*, and *situation-awareness*.

3 Case Study from Transportation

The case study used in this paper is based on an industrial research and development project from transport and logistics industries. One of the world's largest truck manufacturers is developing new transport related services based on an integration and orchestrated interpretation of different information sources, like on-board vehicle information systems, traffic control systems and fleet management systems. Our case aims at using wireless sensor networks in trailers for innovative applications. In comparison to the well-equipped trucks, most of today's trailers are poorly equipped with electronic systems, although they "carry" the actual goods. Trailers are during a transportation assignment often switched between trucks and logistics operators, and they outnumber the number of trucks by far.

The wireless sensor network is installed in the position lights of a trailer. Each position light carries a sensor node able to communicate by ZigBee² with neighboring nodes and equipped with a radar sensor. The radar sensor could be used for protecting the goods loaded on the trailer against theft, offering additional assistance to the driver of the truck (e.g. lane control, blind spot support) or for surveillance of the goods (e.g. sealing different compartments of the trailer). The wireless sensor network in the position lights is controlled by a gateway in the trailer, which communicates with the

² http://www.zigbee.org

back-office of the owner of the trailer or the owner of the goods, and – for some application cases – with the on-board computer of the truck.

Several use cases were defined within the project, which aim at specifying the planned information logistics services for the customer. One of these use cases is a service which contributes to protecting the goods loaded on the trailer against theft. More precisely, the main doors of the trailer are equipped with an additional "electronic" seal. An analysis of current work procedure in the case study showed that when transporting expensive goods, the sending unit of a hauler mounts a physical seal on the trailer's doors and takes a picture of this seal. At the destination, the receiving unit checks whether the seal is broken and compares it with the picture taken at the destination. If the seal is unharmed and looks the same as in the picture, checking the received goods on the trailer can be done less intensely. However, the sealing and picture transmission process as such is time consuming and error prone, which would be improved with an electronic seal. A modified work procedure with electronic seal would look as follows:

- The electronic seal protection service is booked by the trailer owner.
- The goods are loaded on the trailer, doors closed, and seal device is activated, which also activate the protection mode for the trailer.
- At arrival, the responsible person (e.g. a warehouse manager or the driver) sends the "unlock" request.
- If the authorization process for the responsible person is successful (i.e. identity is proven and trailer owner has authorized the person) and the person is in the close vicinity of the trailer, the electronic seal is de-activated.

In case the door is opened with the seal activated, a notification is sent to the backoffice operator who decides on alarming the police or taking other counter-measures.

In order to implement the above services, various kinds of knowledge need to be available and combined, i.e. part of a knowledge base underlying the services. Within the knowledge base observations acquired through the different sensors in the trailer have to be combined with information coming from other sources, like an authentication service for the driver's identity. Furthermore, we have to detect potential critical events, according to what is specified by the IT services. Thus, "context" includes both all characteristics needed to determine the situation of a trailer and the characteristics of the actual information logistics service to be supported. For this purpose, the knowledge base had to accommodate basic transportation domain knowledge, the sensors and their observation possibilities, and a conceptual model for situations.

In addition to the above IT service, many more new services are under preparation. Examples are an electronic fence implemented by radar sensors in the side-marking lights against theft of goods on the trailer, or temperature supervision of cooled cargo on the trailer implemented by temperature sensors spontaneously connecting to the wireless sensor network.

4 Development of the Ontology for Trailer Surveillance (OTS)

In this section we describe the development of a knowledge base represented by the Ontology for Trailer Surveillance (OTS) for the transportation use case presented in section 3. The development process follows and extends the methodology described in [3]. In this section, we first motivate the basics of the OTS and then construct the knowledge base that provides the required features.

4.1. Basics of the Ontology for Trailer Surveillance

As discussed in section 3, the ontology needs to be able to capture knowledge about sensors, situations and the application domain of transportation as such. In this section different information models in sensors, observations, situation (awareness) and time domains are introduced. Utilizing the reusable components of these models the domain model should be able to conceptualize the knowledge base for offering services in transportation sector. Moreover it should serve a basis to prepare a non-exhaustive list of important terms for the particular domains, which could be used as classes and/ or properties.

OTS adopts the Semantic Web Rules Language (SWRL) for modelling rules. SWRL has been proposed as the basic rules language for the Semantic Web Stack and is based on a combination of the OWL DL and OWL Lite with the Rule Markup Language (Rule ML)³. It provides the ability to add Horn-like rules expressed in terms of OWL concepts in order to establish more powerful deductive reasoning capabilities [6], [8]. Observing the relations between objects or entities, *situation awareness* (or assessment) aims at providing a projection based on situations, which describe a state of affairs adhering to a partial view of the world [30]. The three levels of the situation awareness according to [12] are i) perception of elements ii) comprehending the meaning of these elements iii) using the understanding to implicate future states. [9] emphasizes the notion of relationship; the relations between subjects constellate various situations. Whether these subjects are objects from the real world or abstract information objects that are perceived through observations and stored as "facts" in the knowledge base remains undecided. A subject is aware, if he is capable of observing some objects and making inferences from these observations.

Another part of the domain model covers the *sensors* in the trailers and the control hierarchy, which at least consists of the sensor nodes, the trailer gateways, the trailer fleet of a customer of a service type, and the set of all customers of a IT service type. For the trailer-WSN related part of the domain model, The Open Geospatial Consortium (OGC)⁴ sensor web enablement, in particular the observations and measurements (O&M) [1], was taken as starting point. This standard describes conceptual models and defines XML schemas for observations.

The OpenGIS Sensor Model Language Encoding Standard (SensorML) specifies models and XML encoding that provides a framework within the characteristics of sensors. Due to its criticism for complexity, SensorML is not directly adapted in this

³ <u>http://www.w3.org/Submission/SWRL/#1</u> (August 2012)

⁴ http://www.opengeospatial.org/

work. Instead the Starfish Fungus Language (*FL) is utilized, which supports every type of sensor and allows expressing all details about the sensing procedures [5]. Moreover for the modeling of the various sensor types in future the compatibility with SensorML is assured. Last but not least, Sensor Observation Service (SOS) standard defines a Web Service interface which allows querying observations, sensor metadata as well as representations of observed features using three main operations; GetCapabilities, DescribeSensor and GetObservation. In this respective, concepts from an *observation* ontology, Semantic Sensor Observations [15]. The knowledge base, provided by an ontology, can be accessed through a standard SOS request (e.g. GetRequest), making the sensor data useful for a wide range of applications, thus leading to improved interoperability.

OWL allows data values to be typed as XML Schema dates, times or durations and provides minimal support for modelling the temporal relations as well as temporal information. As a result, ontologies often cannot fully express the temporal knowledge needed by applications, forcing users and developers to develop ad hoc solutions. For this purposes the OTS adopts Allen's time intervals algebra that has six basic time intervals constituting a sum of 13 temporal interval relations [17]. On top of this, the valid-time temporal model is applied [16], which attempts at a solution for representing the time information by providing a lightweight temporal model. The selected approaches as well as their application domains are illustrated in Table 1.

Domain	Selected Approaches
Modelling Rules	SWRL
Modelling Time	Allen's Model
Information	Valid Time Model
Modelling Sensors and	OGC Standards
Observations	SemSOS
Modelling Situations	Situation Awareness

Table 1: Modelling domains and selected approaches

4.2. The Ontology for Trailer Surveillance

The OTS should cover the transportation domain with a primary focus on the surveillance of the transportation instances at ground (haulage), i.e. trucks and trailers. The main reason behind using the OTS is offering flexible customer services to protect the transport instances from thievery as described in section 3. In order to specify the requirements on the ontology, we put together a list of competency questions. These are systematized in accordance with their abstraction level (i.e. domain-level or application-level questions) and corresponding architecture (i.e. Observation, Sensor, Event, Situation). Some of those questions are listed in Table 2.

Table 2: Competency questions and their classification

Architecture	Abstraction Level						
	Domain-Level			Application Level			
Observation	Which	observations	are	Give	me	the	observations

	propagated from a feature of				
	interest?	particular trailer instance			
Sensor	Which sensors provide the observations?	Which sensor instances provide information about			
		the velocity?			
Event	Which events are captured from the features?	Is trailer 1 in a safe location?			
Situation	What is the temporal property of a particular situation?	When was the e-seal of trailer1 broken?			

Important Terms and Classes in OTS. The terms utilized in the knowledge base should semantically be explained in order to create a basic terminology and a common understanding among the users as well. Based on the model presented in [18], we define an *event* as concepts, which are caused by observations and aggregated by situations. Events are not moments but they capture the times of the relevant occurrences, such as velocity of a trailer or the distance between the rear doors. Hence one event can occur during another event, which provides useful information for the inference of the instance's situation. Signal assessments are saved as observations in the knowledge base and they all have some values (results). Feature is representation or the abstraction of the real world entity that exists in physical reality [19]. Phenomenon is a physical property that can be observed and measured, such as temperature, gravity [21] . Observation, act of observing a property, produces a result, whose value is an estimate of a property of the observation target or feature of interest [20]. A sensor is a source producing a value within a value space Finally, a situation is a constellation of events over a period of time that affects future system behaviour [18]. Adopting the approach of Baumgartner et al. the situations are described in terms of rule-based situation types comprising objects and the relations between them [13]. These concepts are represented as classes in the ontology, which are depicted in Fig. 1.

The situation classes illustrated in Fig. 1 define and implement the customer services. Hence they are the most important classes in the OTS. It has six defined subclasses four classes are in conformity with the four services that are currently offered to the customers. As an example ESealBroken class represents the implementation of the "Electronic Seal" customer service. In order to assess relevant situations for this service, sensory information has to be aggregated from the individuals of the NonSafeLocationEvent, DistanceEvent and VelocityEvent. The instances of the latter two classes need to occur during some ValidTimeEvent. To name the other important classes, the Entity class represents temporal information based on [16], the Feature class represents the abstraction of real world entities like trailers and platforms, which deploy instances of Sensor class.

Properties of the Classes in OTS. The classes alone cannot provide enough information in an ontology, the properties of these classes are also necessary to constitute the OTS. Due to simplicity and place reasons, only some of the properties should be introduced in this section. The object properties "before, during, equal, meets" are applied for the representation of the time relation following Allen's temporal intervals. The object property deliversIn is used to capture information about the trailers that deliver the goods in particular cities, which are entered manually by the trailer or goods owner to the information base. If a trailer is charged with a delivery in a specific city, then this city is the member of the

SafeCity class. The metadata information of the sensors are represented via hasMetaData object property. The sensory information is interpreted as an observation and this has some values, which are captured through hasResult object property. Unlike object properties, which link individuals to individuals, data type properties describe relationships between individuals and data values. To represent the time information in intervals, hasBegin-hasFinish data type properties are utilized. The data type property hasEnvironment has the value true, if an object is in the vicinity of the trailer.

Rules in OTS. The rules are mainly created to provide consistent time representation such as "if an event meets a second event, which in turn meets a third event, then the first event is before the third event". There are also rules to contribute to the



Fig. 1. Class hierarchy in OTS

consistency of the ontology; for instance, the following simple rule assures that if a situation aggregates an event, then the feature that the event deals with has to be in this situation, since events are captured from features.

The defined classes are classes that have necessary and sufficient conditions. As the name implies such classes have a definition. Classes, all of whose individuals satisfy this definition, can be inferred to be subclasses of a defined class. In the OTS, the concept of the defined classes is used for the subclasses of the Event and Situation. As an example, if the following three conditions are fulfilled, then an individual of the DistanceEvent class is found, i.e. an event happens which could lead to reasoning activities that trigger relevant situations and related to some services: (i) The individual is a member of the event class that are caused by at least one observation and (ii) if such an observation exists, then it must have at least one result and (iii) if such a result exists, then it must have at least one hasDistance data type property with an integer value greater than "1".

These conditions (i) and (ii) are named as "pattern conditions" since most of the defined classes reuse, extend and build upon them. For instance an individual of the ESealBroken class is found if the following conditions are fulfilled⁵: (i) The individual is a member of the situation class that aggregates at least one individual of the NonSafeLocationEvent (ii) The individual is a member of the situation class that aggregates at least one individual of the DistanceEvent and (iii) if such an individual of the DistanceEvent (iv) the individual is a member of the situation class that aggregates at least one individual of the VelocityEvent and (v) if such an individual of the VelocityEvent class exists, then it must happen during at least one ValidTimeEvent.

5 Potentials and Limits of OTS for Self-Organization

The development of OTS primarily followed the requirements indicated by the industrial case in section 3 which did not explicitly include the feature of self-organization. However, the initial experiences with the architecture and new plans to implement adaptability in business models [18] indicated that the ability to adapt to changes in the environment would be of much use. Thus, we will discuss in this section which options exist to use OTS in a self-organizing context.

First, we have to be aware that OTS is based on a multi-tier or multilayer information system architecture. On the technical layer there is a network of wireless sensors that provides basic communication and processing functionality based on self-organisation. This layer is not covered by the OTS and thus it is not reflected which properties the sensor has to have to be an agent. It describes the domain of interest, hence necessary concepts of trailer surveillance. Application logic is based on OTS or in the case of rules even specified in OTS. However, the application tier itself is a multi-layer construct (layers: Sensor Data – Event –Situation – Business Service) and

⁵ The event classes have to fulfill "pattern conditions" already.

is subject to self-organization. Situations for example can be recognized in a decentralized manner by the cooperation of a trailer's sensor nodes.

The discussion will be based on both i) the elements of self-organizing systems: *environment, agents, mechanisms, artefacts*; and ii) the functionalities of self-organizing systems: capability to spontaneously create impromptu network, assemble the network themselves, dynamically adapt to device failure and degradation, manage movement of sensor nodes, and react to changes in task and network requirements (see section 2.3 for reference).

5.1. Coverage of Elements of Self-organizing Systems

An ontology that provides complete support for self-organization needs to provide concepts for all elements of such a system. In the following, we discuss to what extend OTS covers each of the system elements.

There is a broad range of interpretations what has to be considered as the environment of a self-organizing system. It starts from execution *environment* of a software and ranges to physical phenomena in the proximity of an agent or sensor respectively. OTS covers both ends of that scale. The class SensorGrounding represents a certain sensor platform in the sense of used hardware and software. The class Feature and its subclasses represent physical objects in the environment. The PhysicalProperty class describes the data that is covered from the environment by Observations. The assignment to particular features is done by the hasProperty relation.

The *agents* of the self-organizing system are represented by the class Sensor. However, there is no possibility to describe the functionality of the agents besides sensing data. Hence, the only task of an *agent* would be providing Observations. The task of data processing is not covered and cannot be self-organized based on OTS.

Mechanisms in OTS are defined as SWRL-rules. These describe how Observations have to be aggregated to complex interpretations of the environment. This includes the required PhysicalProperties of Features and their aggregation to Events and Situations. Again, the organization of the task of rule interpretation (data processing) is not covered.

Artefacts in the sense of the definition in section 2.3 are represented by instances in the OTS knowledge base.

As a conclusion regarding the coverage of elements of self-organizing system by OTS, it can be said that all elements are addressed. However, there are no *mechanisms* for the organization of data processing. Regarding the discussion at the beginning of this section, this is done on the technical layer. But this task should be performed situation based and content aware. This means, there must be an interface in order to link data interpretation rules and discovered situations to the mechanisms of data processing management, e.g. task assignment.

5.2. Coverage of Necessary Self-organization Capabilities

All mentioned capabilities are necessary for the Technical Layer in order to provide basic communication and processing functionality. However, we focus on the layers that are covered by OTS and discuss, how the ontology provides the knowledge needed for capability provision.

The capability to *spontaneously create impromptu network* is related to the basic task of providing communication functionality. Regarding the multi-layer architecture, this functionality can be clearly assigned to the Technical Layer. OTS layers are not relevant.

The capability to *assemble the network* refers to *mechanisms* for the determination of necessary network components (*agents*) in order to fulfil a certain task. The identification of the right *Agents* for the determination of Events and Situations has to be done in the layers covered by OTS. The OTS rules describe which data from which Sensors (*agents*) is necessary in order to do that. Thus, OTS generally contains the necessary knowledge for the provision of the capability to assemble the network. However, the task of data processing is not covered, as discussed in the previous section.

The capability to *dynamically adapt to device failure and degradation* includes mechanisms for the avoidance of inconsistent states or incorrect data respectively and for the spontaneous construction for workarounds or fall-backs. Regarding OTS, the rules guarantee that Events and Situations are only determined if the complete set of necessary valid data is available. Thus, in the case of a sensor failure the Situations that depend on the respective sensor data cannot be recognized accidently. However, functionality is limited in these cases. OTS does not contain rules that apply for the case of failures and provide for example fall-backs. Such rules cannot simply be added because there is no rule for the non-existence (failure) of an instance. Thus, the addition of failure into the OTS concepts is a prerequisite in order to provide appropriate adaption capabilities to failure and degradation.

The capability to *manage the movement of sensor nodes / agents* implies the reassignment of tasks depending on the current positions of the *agents*. OTS covers the positions of the Sensors relative to objects of the *environment*, e.g. Platform and Trailer. The rules are defined based on these positions. Thus, reassignment of the sensing tasks on position changes is assured.

The capability to *react to changes in task and network requirements* needs *mechanisms* for the reassignment of *agents*' tasks depending on tasks that have to be fulfilled by the system. In OTS, the systems' tasks are described by rules and by instances of the CustomerService class. However, OTS performs all specified tasks for all trailer instances in its current state. There aren't concepts for a more detailed task assignment. Thus, reaction on task changes is only possible on a global level controlled be the (non-)existence of rules and instances of the CustomerService class.

6 Summary and Conclusion

Starting point of this work was the goal to develop an ontology that provides new information logistics services in the transportation sector and that is able to support self-organisation in order to adapt to new situations and requirements. The introduced OTS ontology supports the delivery of already specified new information logistics services like Electronic Seal or Electronic Fence. However, new services can emerge in the future, which require the assessment of different situations. For instance, the ElementarySituation class has no direct function in the OTS whereas it might be used in the future to exploit customer's preparedness to pay for the services, e.g. booking an elementary situation can be provided at a lower price than booking a complex situation, which is represented by ComplexSituation class. Such services can be realized by adding more rules to the knowledge base. New sensor types and situation types will be added by the creation of new instances of the respective classes. The practical evaluation of the OTS has been conducted by adding four trailer instances to the knowledge base, each having different situations and time stamps. In doing so, we were able to observe how well the inference rules work. The future work might include the application of the ontology in a concrete environment.

Developing the ontology revealed the importance of the definition of rules for ontology driven applications. Thus, we added an additional step for rule definition in the ontology development process by Noy and McGuinness [3]. Furthermore, their approach was shifted from the slot-based ontology design to an OWL2 compatible way of ontology creation.

Regarding self-organisation, we conclude that some aspects of self-organization are already well covered by OTS. However, there are also some shortcomings that need to be solved in order to fully support self-organization. A problem is the content aware communication and data processing as proposed for wireless sensor networks. A link between necessary knowledge in order to perform tasks on the upper layers to the processes on the Technical Layer is missing. Additionally, the definition of fall-backs and alternative procedures is missing in OTS and a more comprehensive way of representing service requirements would be desirable. Solving these issues would foster the use of ontologies like OTS for self-organizing information systems.

References

- Cox S (2013) Geographic information Observations and measurements V2.0. OGC Abstract Specification. http://portal.opengeospatial.org/files/?artifact_id=41579. Accessed 14 Feb 2014
- Studer R, Benjamins V.R., Fensel D. Knowledge engineering: Principles and methods. Data & Knowledge Engineering, 25(1–2):161–197, 1998.
- 3. F. Noy N, L. Mcguinness D. Ontology development 101: A guide to creating your first ontology. Development, 32(1):1–25, 2000.
- 4. Öhgren A, Sandkuhl K, editors. Towards a methodology for ontology development in small and medium-sized enterprises, 2005.

- Corcho Ó, Fernández-López M, Gómez-Pérez, A. Methodologies, tools and languages for building ontologies: Where is their meeting point? Data Knowledge Engineering, 46(1):41– 64, 2003.
- De Mola, F., and Quitadamo, R. (2006). Towards an Agent Model for Future Autonomic Communications, *Proceedings of the 7th WOA 2006 Workshop From Objects to Agents*, Catania, Italy, Sept. 26-27, 2006, Retrieved June 18, 2012, from: http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-204/P07.pdf.
- Elmenreich, Wilfried; Meer, Hermann de (2008): Self-Organizing Networked Systems for Technical Applications: A Discussion on Open Issues. In KarinAnna Hummel, JamesP.G Sterbenz (Eds.): Self-Organizing Systems, vol. 5343: Springer Berlin Heidelberg (Lecture Notes in Computer Science), pp. 1–9. Available online at http://dx.doi.org/10.1007/978-3-540-92157-8_1.
- 8. Parsia B, Sirin E, Grau BC et al. (2005) Cautiously Approaching SWRL. http://www.mindswap.org/papers/CautiousSWRL.pdf
- Kokar MM, Matheus CJ, Baclawski K (2009) Ontology-based Situation Awareness. Inf. Fusion 10(1): 83–98. doi: 10.1016/j.inffus.2007.01.004
- Lundqvist, M., 2007. Information Demand and Use: Improving Information Flow within Small-scale Business Contexts, Licentiate of Engineering, Department of Computer and Information Science, Linköpings universitet.
- Lundqvist, M.; Sandkuhl, K.; Seigerroth, U. (2011) Modelling Information Demand in an Enterprise Context: Method, Notation and Lessons Learned. International Journal Systems Modeling and Design, Vol. 2 (3), IGI Publishing, pp. 74-96, 2011.
- Endsley MR (1995) Toward a Theory of Situation Awareness in Dynamic Systems. hum factors 37(1): 32–64. doi: 10.1518/001872095779049543
- Baumgartner N, Gottesheim W, Mitsch S et al. (2010) Editorial: BeAware!-Situation Awareness, the Ontology-driven Way. Data Knowl. Eng. 69(11): 1181–1193. doi: 10.1016/j.datak.2010.07.008
- Kurt Sandkuhl: Information Logistics in Networked Organizations: Selected Concepts and Applications. Enterprise Information Systems, 9th International Conference, ICEIS 2008. LNBIP, Springer.
- Henson CA, Pschorr JK, Sheth AP et al. (2009) SemSOS: Semantic Sensor Observation Service. In: Proceedings of the 2009 International Symposium on Collaborative Technologies and Systems. IEEE Computer Society, Washington, DC, USA, pp 44–53
- Holzer, Richard; Meer, Hermann de; Bettstetter, Christian (2008): On Autonomy and Emergence in Self-Organizing Systems. In KarinAnna Hummel, JamesP.G Sterbenz (Eds.): Self-Organizing Systems, vol. 5343: Springer Berlin Heidelberg (Lecture Notes in Computer Science), pp. 157–169. Available online at http://dx.doi.org/10.1007/978-3-540-92157-8_14.
- 17. Hemalatha M, Uma V, Aghila G (2012) Time ontology with Reference Event based Temporal Relations (RETR). International Journal of Web & Semantic Technology 3(1)
- Sandkuhl, Kurt; Borchardt, Ulrike; Lantow, Birger; Stamer, Dirk; Wißotzki, Matthias (2012): Towards Adaptive Business Models for Intelligent Information Logistics in Transportation. In Natalia Aseeva, Eduard Babkin, Oleg Kozyrev (Eds.): Perspectives in Business Informatics Research - 11th International Conference, BIR 2012, Nizhny Novgorod, Russia, September 24-26, 2012. Satellite Workshops and Doctoral Consortium. Nizhny Novgorod.

- 19. Probst F (2007) Semantic Reference Systems for Observations and Measurements. PhD, University of Münster
- 20. Bröring A, Stasch C, Echterhoff J (2012) Sensor Observation Interface Standard v2.0. https://portal.opengeospatial.org/files/?artifact_id=47599
- 21. Botts M, Robin A (2007) OpenGIS® Sensor Model Language SensorML Implementation Specification