Birger Lantow
Vladimir Tarasov
(Eds.)

2nd International Workshop on Ontologies and Information Systems

13th International Conference on Perspectives in Business Informatics Research
Lund, Sweden, September 22-24, 2014
Workshop Proceedings

Universität Rostock Traditio et Innovatio

JÖNKÖPING UNIVERSITY

Title
2nd International Workshop on Ontologies and Information Systems

Sub-title
13th International Conference on Perspectives in Business Informatics Research
Lund, Sweden, September 22-24, 2014
Workshop Proceedings

Volume Editors
Birger Lantow
The University of Rostock
Institute of Computer Science
Albert-Einstein-Str. 22
18057 Rostock, Germany
E-mail: birger.lantow@uni-rostock.de

Vladimir Tarasov
Jönköping University
School of Engineering
Box 1026
55 111 Jönköping, Sweden
E-mail: vladimir.tarasov@jth.hj.se

Preface

Ontologies are a knowledge representation technique of growing importance - not only for intelligent information supply but equally for information systems and industrial applications. In conjunction with evolving semantic technologies for ontology engineering, representation and query, new ways open up for ontologies and information systems (IS) integration, combination and use. In the structural perspective, ontologies can provide means to structure, store and access generic IS content. In the temporal perspective, ontologies can guide the development of new IS. They may help to choose appropriate processes, algorithms, rules, and software components depending on the requirements.

WOIS 2014 — the 2nd International Workshop on Ontologies and Information Systems — had the aim to bring together people who have a strong interest in the innovative use of these technologies and approaches in the context of enterprises and public organizations. The workshop took place on September 22, 2014, in Lund (Sweden) as part of the 13th International Conference on Perspectives in Business Informatics Research (BIR 2014). Based on at least three reviews per submission the international Program Committee selected 6 high-quality papers for inclusion in this volume. The authors of these papers include both researchers and practitioners from different disciplines. The WOIS 2014 program reflects different facets of the workshop topics, including organizational and social issues, as well as methodical and technical aspects related to the use of ontologies in information systems lifecycle.

We dedicate special thanks to the members of the international Program Committee for promoting the workshop, their support in attracting high-quality submissions, and for providing excellent reviews of the submissions. Without their committed work a high-quality workshop like WOIS 2014 would not have been possible. Our thanks also include the external reviewers supporting the paper selection process.

September, 2014

Birger Lantow
Vladimir Tarasov

Organization

WOIS 2014 Chairs

Birger Lantow	University of Rostock, Germany
Vladimir Tarasov	University of Jönköping, Sweden

WOIS 2014 Program Committee

He Tan, Jönköping School of Engineering (Sweden)

Karl Hammar, Jönköping School of Engineering (Sweden)

Kurt Sandkuhl, University of Rostock (Germany)

Tatiana Levashova, St. Petersburg Institute Inform. & Autom. (Russia)

Nikolay Shilov, St. Petersburg Institute Inform. & Autom. (Russia)

Ralf D. Kutsche, TU Berlin (Germany)

Eduard Babkin, Higher School of Economics Nizhny Novgorod (Russia)

Marite Kirikova, Riga Technical University (Latvia)

Table of Contents

Ontology-based Detection of States and Situations

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 years[1], 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 self-organization: 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 ZigBee[2] 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 back-office 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] http://www.w3.org/Submission/SWRL/#1 (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 Observation Service (SemSOS or O&M-OWL), are adopted, which takes the advantages of representing the sensor data in OWL and enabling reasoning over 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.

Table 1: Modelling domains and selected approaches

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

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?	which are assessed from a 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

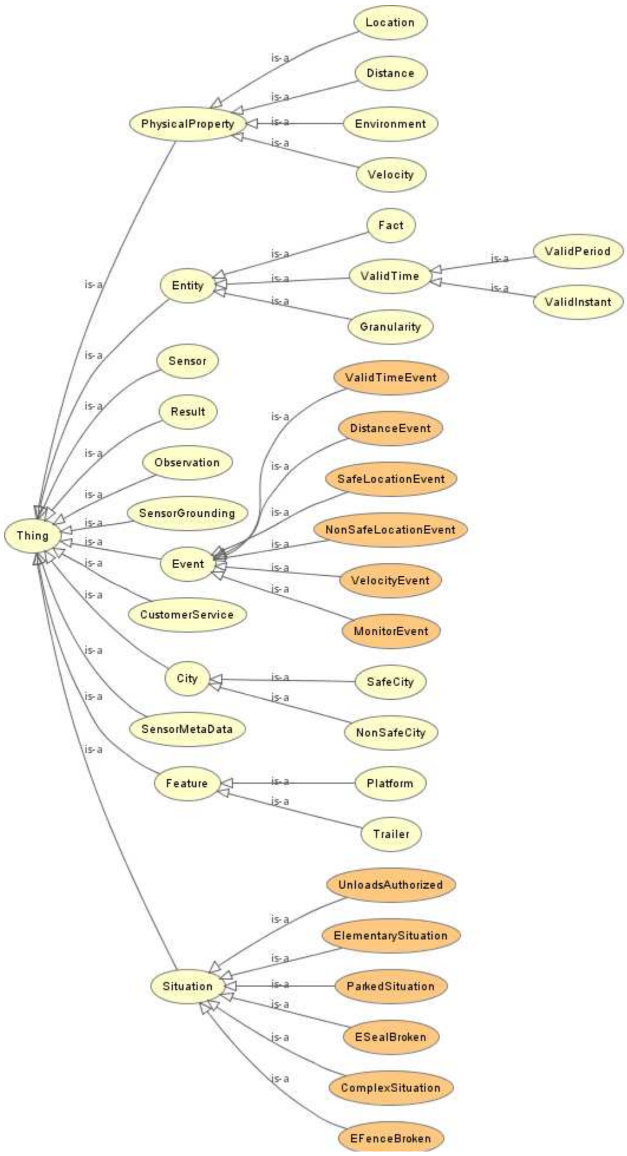


Fig. 1. Class hierarchy in OTS

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 fulfilled[5]: (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` class exists, then it must happen during at least one `ValidTimeEvent` (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 `Observation`s. 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 `Observation`s. 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 `Observation`s have to be aggregated to complex interpretations of the environment. This includes the required `PhysicalPropertie`s of `Feature`s and their aggregation to `Event`s and `Situation`s. 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 *mechanism*s 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 `Event`s and `Situation`s has to be done in the layers covered by OTS. The OTS rules describe which data from which `Sensor`s (*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 `Event`s and `Situation`s are only determined if the complete set of necessary valid data is available. Thus, in the case of a sensor failure the `Situation`s 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 `Sensor`s 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

1. 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
2. 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.

5. 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.
6. 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.
7. 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
9. 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
10. 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.
11. 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.
12. Endsley MR (1995) Toward a Theory of Situation Awareness in Dynamic Systems. hum factors 37(1): 32–64. doi: 10.1518/001872095779049543
13. 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
14. Kurt Sandkuhl: Information Logistics in Networked Organizations: Selected Concepts and Applications. Enterprise Information Systems, 9th International Conference, ICEIS 2008. LNBIP, Springer.
15. 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
16. 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)
18. 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

Situation Detection Based on Knowledge Fusion Patterns

Tatiana Levashova[2], Kurt Sandkuhl[1,3], Nikolay Shilov[2,3], Alexander Smirnov[2,3]

[1]The University of Rostock, Institute of Computer Science
Chair Business Information Systems, Albert-Einstein-Str. 22, 18059 Rostock, Germany
Kurt.Sandkuhl@uni-rostock.de

[2]St.Petersburg Institute for Informatics and Automation of the
Russian Academy of Sciences, 39, 14 Line,
199178 St. Petersburg, Russia

[3]ITMO University,
49, Kronverkskiy pr., 197101
St. Petersburg, Russia,

Abstract. With the increasing use of sensors and actuators in technical systems and knowledge-intensive services the need for processing the information captured by these sensors and "making sense" out of it increases. Knowledge fusion is supposed to contribute to this field since it aims at integrating knowledge from different sources. Development of knowledge fusion solutions is a complex task which can be compared to systems and software development. As in other development areas there is a need for efficient development processes which can be supported by reusing solution parts, such as patterns or components. The paper brings together experiences from knowledge fusion sub-system development and from design of knowledge fusion patterns. The main contributions of this paper are (1) a real-world application scenario presenting typical requirements to knowledge fusion systems, (2) application of knowledge fusion patterns from context-based decision support to situation recognition, (3) recommendations from this application case.

Keywords: knowledge fusion, knowledge fusion pattern, situation, situation detection, knowledge logistics.

1 Introduction

With the increasing use of sensors and actuators in technical systems and knowledge-intensive services, like in cyber-physical systems, preventive maintenance or intelligent information logistics, the need for processing the information captured by these sensors and "making sense" out of it increases. Knowledge fusion is supposed to contribute to this field since it aims at integrating knowledge from different sources.

The development of knowledge fusion solutions and systems usually is a complex task which can be compared to systems and software development projects. As in other development areas there is a need for efficient development processes which can be supported by reusing solution parts, such as patterns or components. The aim of this paper is to bring together experiences from knowledge fusion sub-system development and from design of knowledge fusion patterns. We will analyze applicability and pertinence of knowledge fusion patterns in a past project from civil security and derive recommendations from this analysis for future projects aiming at using fusion patterns.

The main contributions of this paper are (1) a real-world application scenario presenting typical requirements to knowledge fusion systems, (2) application of knowledge fusion patterns from context-based decision support to situation recognition, (3) recommendations from this application case.

The remaining part of the paper is structured as follows: section 2 gives an overview to the field of knowledge fusion and discusses related work. Section 3 presents the application case constituting the frame for this research. An overview to the basic concept of knowledge fusion pattern is given in section 4. Section 5 discusses the applicability of fusion patterns in the given application case and derives recommendations. Conclusions and future work are discussed in section 6.

2 Knowledge Fusion

Techniques for data, information and knowledge fusion from different sensors, services and components have received much attention during the last decade. This section will give a brief overview to the field which starts from data fusion since this often lays the ground for higher level fusion activities, like knowledge fusions.

The process model for data fusion suggested by Joint Directors of Laboratories (JDL) which later became the Data Fusion Group (DFG) is the most popular of the fusion models. First proposed in 1985, the JDL/DFG model was revised several times (see [1] and [2]) due to observed shortcomings [3]. Currently, the levels with the JDL/DFIG model are: Source Pre-processing/Subject Assessment (level 0), Object Assessment (level 1), Situation Assessment (level 2), Impact Assessment / Threat Refinement (level 3), Process Refinement (level 4), and User Refinement / Cognitive Refinement (level 5). Through its different levels, the model divides the processes according to the different levels of abstraction of the data to be fused and the different problems for which data fusion is applicable (e.g. Characteristic estimation vs. situation recognition and analysis). The model doesn't prescribe a strict ordering of the processes and the fusion levels, and the levels are not always discrete and may overlap. The model was initially proposed for the military applications but is now widely used in civil domains as well, such as business or medicine. The JDL/DFIG model is useful for visualizing the data fusion process, facilitating discussion and common understanding and important for systems-level information fusion design [4].

Other fusion models include the Boyd loop [5], the Waterfall model [6] and the Endsley model [7], which focus on different perspective of the fusion task and propose refined structures or processes. The Omnibus model [8] is an attempt to achieve a unified model by merging different fusion models. It reflects the cyclic nature of the Boyd loop, and carries the finer structure of the Waterfall model, of the JDL model, and of the Endsley model.

Two main groups of knowledge fusion approaches are commonly distinguished: knowledge fusion based on knowledge representation technologies and semantic integration for federated systems. Semantic integration will not be discussed in detail, since it primarily focuses on applications in information systems and database context, i.e. fusing schemata of the information sources. Knowledge Fusion approaches based on knowledge representation techniques, like semantic nets or ontologies, have been subject to research during the last 20 years and resulted in a number of methodology and technology approaches. The most cited and used approaches include the following ones.

The KRAFT [9] architecture for knowledge fusion and transformation: knowledge fusion is defined as a combination of knowledge from disparate sources in a highly dynamic way. In order to do this, data instances need to be associated with knowledge concerning their context, such as how they should be interpreted and how they can be used. Two main kinds of operations are recommended in a distributed knowledge fusion system: knowledge retrieval (to find out everything the organization knows about something) and problem solving (to use the combined knowledge to solve a particular problem).This leads to a number of services required by the system: knowledge location services (to find the relevant knowledge on the network), knowledge transformation services (to translate the knowledge into a common representation language), and knowledge fusion services (to combine and process knowledge).

The Knowledge Supply Net approach KSNet [10]: the goal of the KSNet approach is to complement insufficient knowledge and obtain new knowledge using knowledge from different sources. The technologies involved spans from ontology management and intelligent agents to constraint satisfaction and soft computing. Knowledge as a set of relations, such as constraints, functions, or rules, that can be used by a user or expert in order to decide how, why, where, and what to do with the information in order to meet a goal or a set of goals within a clear context and time. The knowledge fusion process structure has several steps, including translating knowledge from different knowledge sources into a unified form, acquiring knowledge from external sources, select the relevant knowledge producing new knowledge by discovering or deriving it from the existing knowledge, internalization of knowledge, and knowledge fusion management.

The general idea to capture the domain under consideration in a domain ontology, the tasks to be supported in task ontologies being a part of the domain ontology, integration the knowledge sources by using these ontologies and fusing the relevant knowledge on-demand is suitable for the planned project. A commonality between these approaches is that industrial scale application and support by off-the-shelf products so far is quite sparse.

3 Application Scenario

The content of this paper is based on work from the FP7-Security-IP Integrated Mobile Security Kit (IMSK) project[6]. IMSK was addressing the continuously evolving threat of unpredictable terrorist activity, which demands the application of existing and developing technology for the protection of citizens. More concretely, IMSK combines technologies for area surveillance, checkpoint control, CBRNE detection and support for VIP protection, into a mobile system for rapid deployment at venues and sites which temporarily need enhanced security. The project's approach is to design a system (IMSK) that will integrate heterogeneous information to provide a common operational picture. This includes to employ legacy and novel sensor technologies, and to adapt the system to local security forces.

Data, information and knowledge fusion have major roles within the IMSK system. IMSK integrates different kinds of sensors providing observations of the sites to be protected. The data provided by physical sensors, as well as pieces of information provided by human observers and open sources, have to be combined in order to provide an overview of the ongoing situation. Within this work, we are particularly interested in knowledge fusion and fusion of high level information. We decompose the fusion process into several phases of fusion. First, entities of the world are represented using detailed observations provided by different kinds of sensors. This is the attribute fusion phase. Then, the recognized entities are combined and relations among them are observed. The situation fusion phase aims at reconstructing a more global view of the observed situation that contains both the entities recognized in the attribute fusion phase and the relations that have been observed among them by other information sources providing information of a higher level. Both the attribute and situation fusion, rely on the same approach. The two phases differ only by the level of detail of the observations that are processed. Once the representation of an ongoing situation is achieved, the situation recognition aims at deciding whether the ongoing situation is one of the "critical situations" preliminary defined by the end users. Last, the event correlation phase allows for combining the different static critical situations recognized in order to detect the occurrence of complex critical situations. The event correlation phase allows taking into account time and space issues of the critical event detection process.

Several scenarios were defined within the IMSK project that aim at showing the adaptability of the platform to different types of environments and events. One of these scenarios is the protection of VIPs[7] during an EU summit. The events of the summit take place in three different locations of a city. The participants have thus to go from one place to another one. One of the tasks to be supported is the protection of VIPs when crossing a bridge when going from the congress center to the dinner place. Several sensors are deployed in order to detect CBRN[8] threats, fireworks, approaching vehicles, etc. Our aim, within knowledge fusion, is to combine observations acquired through the different sensors (and potentially already fused at a low level), with

[6] http://www.imsk.eu
[7] VIP = Very Important Person
[8] CBRN = chemical, biological, radiological and nuclear

information coming from other sources. We then have to detect potential critical situations and events, according to the ones that are specified by the end users of the IMSK system. Our example here focuses on the detection of a vehicle approaching a VIP while he/she crosses the bridge. We use the vehicles tracking system observations, the schedule of the summit and observations provided by people on the site.

The requirements to the functionality of a knowledge fusion sub-system derived from this scenario are defined as “capabilities”, i.e. desirable functionalities to be supported by knowledge fusion. Examples for capabilities are:

- Area surveillance: Area control: airspace, Area control: land, Area control: waterways, Protection of public infrastructure, Protection of buildings and Protection of property.
- Command & control (C2): Situation awareness, Decision support, Deployment support and Communication management
- Communications: Emergency communication, secure communication and communication in buildings.
- Access rights: verification of access rights, enforcement of access right restrictions, crowd monitoring, identification if unwanted behavior, VIP assault prevention, identification of wanted people.

4 Knowledge Fusion Patterns

Knowledge fusion patterns were developed to generalize knowledge fusion processes in relation to sources involved in these processes. For this, the knowledge fusion processes ongoing in a context-aware decision support system (CADSS) were investigated [11].

In the CADSS a situation is modeled by a two-level context. Abstract and operational context represent the situation at the first and second levels, respectively.

The abstract context is a non-instantiated ontology-based situation model. This context is created for a specific situation. It captures knowledge relevant to this situation from an application ontology. The ontology combines domain and task knowledge needed to describe situations happening in the application domain.

The operational context is the result of an abstract context instantiation for the actual circumstances. Data and information from various sources (sensors, humans, etc.) is fused within the abstract context structure to produce the operational context. This context is a near real-time schematic picture of the ongoing situation.

The operational context is the basis for decision making. The system supports the decision maker with a set of decisions feasible in the current situation. This set is a result of solving tasks specified in the abstract context as a constraint satisfaction problem.

The investigation of the processes ongoing in the CADSS results in the following knowledge fusion patterns:

Selective fusion: integration of multiple knowledge pieces from various ontologies of different types into a new ontology. The pattern is used for application ontology creation.

Simple fusion: integration of multiple knowledge pieces from a single large multipurpose ontology into a new knowledge piece intended to restricted purposes. The pattern is used for abstract context building.
Extension: inference of new knowledge as a result of knowledge integration. The pattern is used for abstract context building.
Instantiated fusion: fusion of data/information from multiple (possibly, heterogeneous) sources to create a representation that may be used by the CADSS, decision makers, and other humans as the basis for problem solving and decision making.
Flat fusion (see Fig. 1): fusion of knowledge from multiple knowledge sources during problem solving. The pattern is used for generation of a set of feasible decisions.
Adaptation: gaining new capacities/capabilities by units (knowledge sources, source network, actors, etc.) as a result of their adaptation to new circumstances or new scenarios. The pattern is used for adaptation of an existing knowledge source network to new scenarios and for adaptation of decision executives to changing settings.
Historical fusion: revealing new knowledge from hidden knowledge based on the accumulated one. The pattern is used to inductive inference of new relations between the entities presenting in different contexts.

The knowledge fusion patterns are formalized in terms of preservation/change of the structures and autonomies of the initial and target sources, and in terms of the results the knowledge fusion processes produce in the CADSS. The reasons of choice the states for structures and autonomies as a measure are as follows.

Name: flat fusion
Problem: providing the decision maker with a set of alternative decisions
Solution: solving the problems, to which the decision maker has to find solutions in the current situation, as a constraint satisfaction problem
Initial source: operational context
Target source: a knowledge source fusing operational context and the set of alternatives

Autonomy pre-states:	*initial source*	*target source*
	non-autonomous	n/a*

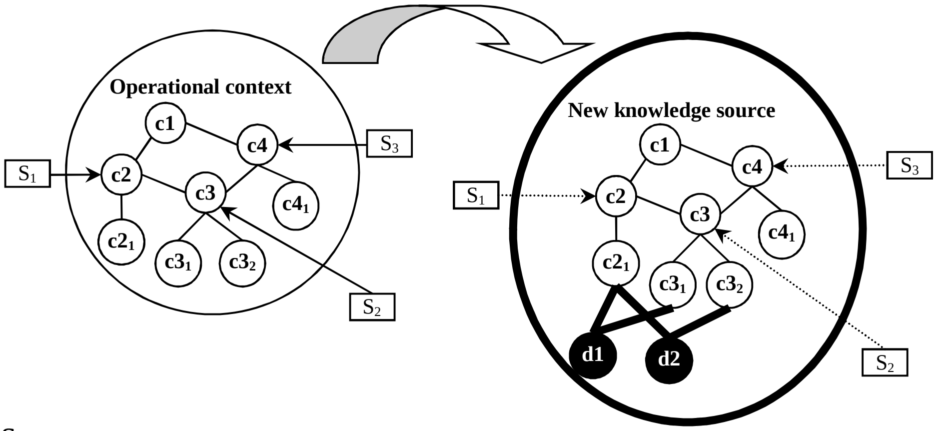
Result in CADSS: a new knowledge source of a new type
Result in ontology terms: a new knowledge source representing the result of fusion of the dynamic ontology with the set of alternative decisions

Post-states:	*initial source*	*target source*
Structure:	changed	n/a
Autonomy:	n/a	autonomous

Schematic representation: Fig. 2
Phase of CADSS functioning: generation of a set of alternative decisions

*n/a means the source does not exist

Fig. 1. Flat fusion



S_n – source
c_n – internal units in representations of sources
d_n – feasible decision
—— – relationship
⟶ – link between the knowledge source and the unit being instantiated
·····➤ – reference to the unit to be instantiated
▬▬ – new knowledge

Fig. 2. Flat fusion: schematic representation

Knowledge fusion involves multiple sources in the integration processes. In the context-aware systems integration of data/information/knowledge refers to the process of integration of their conceptual structures. Therefore, source's structure is an obligatory concept taken into account by the integration.

Autonomy creates awareness of the reliability of data/information/knowledge represented in the sources. The CADSS operates in dynamic environments. Information and knowledge represented in the environmental sources that are related to the internal system sources (i.e., the environmental sources and system ones are non-autonomous) are considered to be more reliable than information/knowledge represented in the autonomous environmental sources. An argument in favor of this is any changes in the linked (non-autonomous) environmental sources are reflected in the system sources.

An example of patterns specification is given in Fig. 1; a schematic representation in Fig. 2. Flat fusion patter is used in this example.

5 Knowledge Fusion Patterns for Situation Detection

Within IMSK, the domain modeling and knowledge representation is based on ontologies [12]. They are used as the core representation paradigm and formalism. The knowledge representation for the fusion module includes two main categories of knowledge: (1) knowledge specifying fusion tasks and (2) knowledge forming the input for these fusion tasks. The fusion tasks to be supported are attribute and observation fusion, critical situation recognition and event correlation. In this paper, we will focus on situation recognition and the attribute and observation fusion forming the basis for it. An ontology-based formalization of the situation model is available in [12].

Within the attribute fusion, the different features acquired through the various sensors of IMSK are combined so to determine the identity of the objects and entities taking part of the external situation. As opposed to kinematic information (i.e. position, velocity and acceleration), attribute information provide descriptive information about an entity's characteristic or quality. The ID-tag, color, width or acoustic signature on an entity all make plausible attributes. Attributes are, by many means, useful within systems such as IMSK. In crowded spaces, attributes can facilitate a tracker to associate observations to correct tracks. A rich set of attributes can also support in the situational- and behavioral analysis, e.g. by determining the identity of entities, by establishing their relations, and by indicating odd attribute combinations.

The aim of the attribute fusion module is to build a more precise and complete description of the entities taking part in an observed situation. This is made by continuously trying to extend and refine the flora of attributes associated with each entity. For this task, we use heterogeneous sensors and take advantage on their different qualities and the kind of attributes they can deliver. During the situation fusion phase, the focus is on the relations that exist between these different entities. Finding these relations allow having a more coherent representation of the ongoing situation. The representation goes from a set of observed entities to a structured observed situation in which the previous entities take part, with specific roles. When two observations (at least partially) overly, the information fusion sub-processes builds an unique view of the observed object or situation from them. The fusion phase confronts several points of view on the state of an object or a situation. This confrontation leads to a conflict resolution phase. A major stake of information fusion is to automate the conflict resolution phase.

Fig. 3 describes the general information flow used for situation recognition. IMSK smart sensors, fusion modules and open information sources provide information which are captured in observation graphs based on the domain model. These graphs are used for observation fusion and create fused observation graphs. The observation graphs are the basis of situation recognition, which essentially is based on comparison with a-priori defined models of critical situations. In case a critical situation is detected, alerts are generated in the command & control system.

When investigating the use of knowledge fusion patterns for situation recognition in IMSK, the following process was used: we first matched the decision support process forming the ground for knowledge fusion patterns onto the situation recognition flow in IMSK. Afterwards the matching phases were investigated in more detail in order to identify potentially suitable patterns. The potentially suitable patterns then were mapped onto the IMSK ontology in order to finally decide on applicability.

The first step, mapping the decision support process on the situation recognition flow, showed a principal difference in the approaches. IMSK did not explicitly use an abstract context which was adapted and configured for the actual operative situation but rather applied the same application ontology, which was configured for the

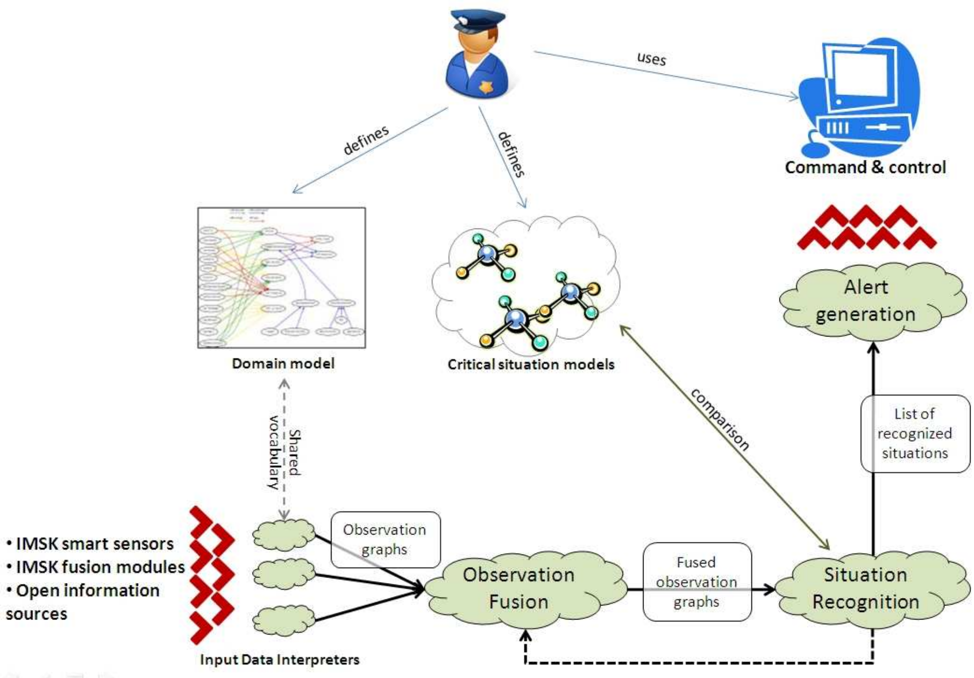


Fig. 3. General information flow for situation recognition

application case under consideration by instantiating it. Although the approaches have similarities, the fundamental difference is that structural changes and extensions as supported by the knowledge fusion patterns addressing the process flow in adapting the abstract context (abstract context creation, refinement and reuse) are not applicable. It should be noted that the use of "context" from decision support systems in IMSK is appropriate, since both event correlation and situation recognition depend on the actual situation of an entity (e.g. the access control sub-systems for the EU-summit).

The knowledge fusion patterns defined for the "operative" part of the decision support system process could be applied even for IMSK. An example is the "flat fusion" pattern presented in section 4. This pattern would be used to create a list of critical situations based on the representation of observations in the knowledge base. In DSS, flat fusion creates a list of feasible solutions for a decision problem. In situation recognition, this "feasible solution" correspond to possible situations.

Other knowledge fusion patterns considered as useful and pertinent for the situation recognition scenario are "instantiated fusion" (for creating a real-time representation of the current situation based on the observations), "historical fusion" (for creating new knowledge based on archived previous critical situations) and "adaptation" (for adaptation of the knowledge base to the detected situation). Fig. 4 gives an overview to the fusion process and usable knowledge fusion patterns in IMSK.

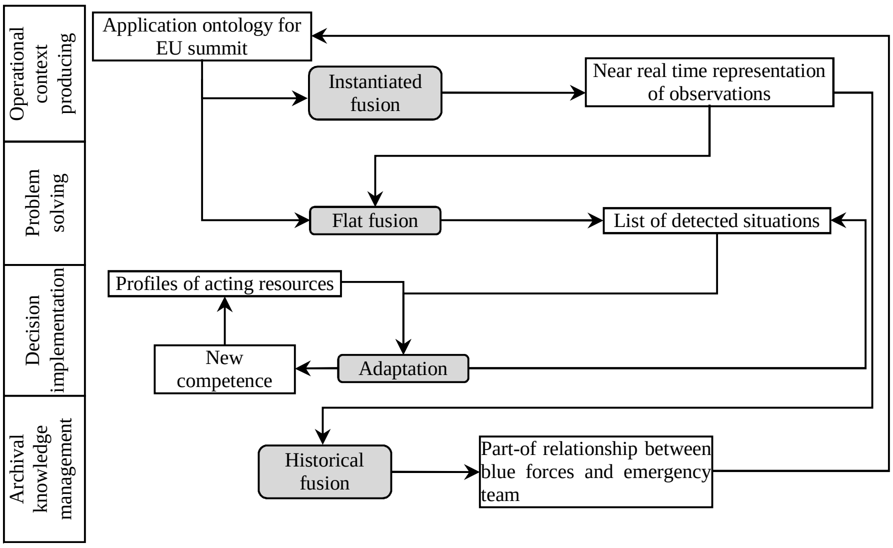


Fig. 4. Usage of knowledge fusion patterns for planning fire response actions

In addition to the use of KF (knowledge fusion) patterns in the knowledge fusion process, we also investigated possible knowledge fusion results [13 – 23]. Potential results enhancing the knowledge base are:

- new knowledge created from data/information. Such knowledge represents information having been processed, organized or structured in a way that may be used by systems and humans as the basis for problem solving and decision making;
- a new type of knowledge. This result means integration of such knowledge that the outcome is knowledge of a radically new type;
- a new knowledge about the conceptual scheme. This result concerns changes in schemes formally representing knowledge. New relations, concepts, properties, etc. appearing in existing schemes are examples of new knowledge;
- a new problem solving method or a new idea how to solve the problem. This is the result of reuse and combining existing knowledge in new scenarios;
- new capabilities/competencies of a unit (a unit that produces or contains knowledge). Like the item above, the new capabilities/competencies are the result of reuse and combining existing knowledge in new scenarios;
- a solution for the problem. This outcome means integration of knowledge from various sources in problem solving, which results to a problem solution;
- a new knowledge source created from multiple sources. This result is a generalization of different knowledge fusion results. It implies origination of a new source to represent the new knowledge.

6 Summary and Future Work

The paper investigated the possibility of KF pattern use in situation recognition using the example of civil security from the IMSK project. The main result was that four KF patterns from the operative part of the knowledge fusion process were found applicable and useful. Based on this insight, we recommend that future projects aiming at the development of knowledge fusion systems for situation recognition should take these patterns into account. We expect this to speed up the construction process of the domain model and the knowledge base. Another result of this investigation is that KF patterns designed for DSS at least on the conceptual level can be transferred to another knowledge fusion purpose: situation recognition.

The main limitation of the research presented here is that it stays on a conceptual level. It would be worthwhile and interesting to develop an actual knowledge fusion solution for situation recognition based on KF patterns. During this development process, the efforts spend would have to be documented and compared to other projects in order to validate whether pattern use really saves efforts.

Acknowledgements

This work has been performed as part of the EU-FP7 funded project Integrated Mobile Security Kit (IMSK) 2009-2013. This work was also partially supported by the Project 213 (the research program “Information, control, and intelligent technologies & systems” of the Russian Academy of Sciences (RAS)), the Project 2.2 (the Nano- & Information Technologies Branch of RAS), projects 13-07-12095, 14-07-00345, 14-07-00427 (the Russian Foundation for Basic Research), and Grant 074-U01 (the Government of Russian Federation).

References

1. Hall, D. L., and Llinas, J. (Eds.) (2001). *Handbook of Multisensor Data Fusion*, CRC Press.
2. Llinas, J., Bowman, C., Rogova, G., Steinberg, A., and White, F. (2004). Revisiting the JDL data fusion model II, Svensson, P., and Schubert, J. (Eds.), Proceedings of the Seventh International Conference on Information Fusion.
3. Hall, D. L. and Steinberg, A. N. (2001). Dirty Secrets in Multisensor Data Fusion, Hall, D.L., and Llinas, J. (Eds.), Handbook of Multisensor Data Fusion, CRC Press, 21-1 – 21-12.
4. Blasch, E.P., E. Bosse, and Lambert, D. A. (2012). High-Level Information Fusion Management and System Design, Artech House Publishers, Norwood, MA.
5. Boyd, J. (1987). A Discourse on Winning and Losing, Maxwell AFB lecture, 1987.
6. Markin, M., Harris, C., Bernhardt, M., Austin, J., Bedworth, M., Greenway, P., Johnston, R., Little A., and Lowe, D. (1997). Technology Foresight on Data Fusion and Data Processing, The Royal Aeronautical Society, 1997.
7. Endsley, M.R. (1995). Toward a Theory of Situation Awareness in Dynamic Systems, Human Factors Journal, 37(1), 32-64.
8. Bedworth, M., and O`Brien, J. (2000). The Omnibus Model: A New Model of Data Fusion? IEEE Aerospace and Electronic Systems Magazine, 15(4), 30-36.
9. Preece, A., Hui, K., Gray, A., Marti, P., Bench-Capon, T., Jones, D., and Cui, A. (1999). The KRAFT architecture for knowledge fusion and transformation. Proceedings of the 19th

SGES International Conference on Knowledge-Based Systems and Applied Artificial Intelligence (ES'99). Springer.

10. Smirnov, A., Pashkin, M., Shilov, N., Levashova, T. (2003). KSNet-Approach to Knowledge Fusion from Distributed Sources, Computing and informatics, 22(2), 2003.
11. Smirnov, A., Levashova, T., Shilov, N.: Patterns for Context-Based Knowledge Fusion in Decision Support. Information Fusion. http://dx.doi.org/10.1016/j.inffus.2013.10.010 (2013).
12. Laudy, C., Petersson H., Sandkuhl, K. (2010) Architecture of Knowledge Fusion within an Integrated Mobile Security Kit. Proc. FUSION 2010, Edinburgh, UK.
13. R. Scherl, D.L. Ulery, Technologies for army knowledge fusion, Final report, Monmouth: Monmouth University, Computer Science Department, West Long Branch; 2004 Sep., Report No. ARL-TR-3279.
14. P. Alun, K. Hui, A. Gray, P. Marti, T. Bench-Capon, Z. Cui, D. Jones, Kraft: an agent architecture for knowledge fusion, Int. J. Coop. Inf. Syst. 10 (1–2) (2001) 171–195.
15. C.W. Holsapple, A.B. Whinston, Building blocks for decision support systems, in: G. Ariav, J. Clifford (Eds.), New Directions for Database Systems, Ablex Publishing Corp., Norwood, 1986, pp. 66–86.
16. K.-R. Lee, Patterns and processes of contemporary technology fusion: the case of intelligent robots, Asian J. Technol. Innov. 15 (2) (2007) 45–65.
17. H.A. Grebla, C.O. Cenan, L. Stanca, Knowledge fusion in academic networks, BRAIN: Broad Res. Artif. Intell. Neurosci. 1 (2) (2010) URL: http://www.edusoft.ro/brain/index.php/ brain/article/download/60/145 (access date: 18.07.2014).
18. T.-T. Kuo, S.-S. Tseng, Y.-T. Lin, Ontology-based knowledge fusion framework using graph partitioning, in P.W.H. Chung, C.J. Hinde, M. Ali (Eds.), IEA/AIE 2003, LNAI 2718 (2003) 11–20.
19. J. Gou, J. Yang, Q. Chen, Evolution and evaluation in knowledge fusion system, in: J. Mira, J.R. Alvarez (Eds.), IWINAC 2005, LNCS 3562, 2005, pp. 192–201.
20. M.J. Roemer, G.J. Kacprzynski, R.F. Orsagh, Assessment of data and knowledge fusion strategies for prognostics and health management, in: Proc. 2001 IEEE Aerospace Conference. Big Sky, Montana, USA, 2001, vol. 6, pp. 2979–2988.
21. K.B. Laskey, P. Costa, T. Janssen, Probabilistic ontologies for knowledge fusion, in: Proc. 2008 11th Intl. Conf. Information Fusion, 2008, Cologne, Germany, IEEE, URL: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp? arnumber=4632375 (access date: 18.07.2014).
22. C. Jonquet, P. LePendu, S. Falconer, A. Coulet, N.F. Noy, M.A. Musen, N.H. Shah, NCBO Resource Index: ontology-based search and mining of biomedical resources, J. Web Semant. 9 (3) (2011) 316–324.
23. L.Y. Lin, Y.J. Lo, Knowledge creation and cooperation between cross-nation R&D institutes, Int. J. Electron. Bus. Manag. 8 (1) (2010) 9–19.

Towards Compliance Checking between Business Process Models and Lawful States of Objects

Ludmila Penicina and Marite Kirikova

Institute of Applied Computer Systems, Riga Technical University, 1 Kalku, Riga, LV-1658, Latvia
{ludmila.penicina, marite.kirikova}@rtu.lv

Abstract. We address the existing gap between business process models and lawful states of business objects. This gap hinders compliance of business process models with internally and externally imposed regulations. Existing modelling methods such as BPMN and ArchiMate lack an explicitly declarative approach for capturing flow of business objects, their states and laws of state transitions. Such deficiency can cost organization potential legal problems, make the ability of BPMN and ArchiMate to capture real-world phenomena questionable and drive modellers to employ additional standards. This paper proposes a formalized solution for closing the gap between business process models and states of business objects by using BWW model. Our approach includes means for explicit definition of states of business objects, automatic generation of conceivable state space at a process model design-time, and automatic generation of lawful state space and compliance checking at a process run-time.

Keywords: Business process modelling, BWW, BPMN, Object state, Compliance.

1 Introduction

Business processes are valuable assets of any organization. In organizations business process modelling has become a main activity for capturing, analysing, and improving business processes. Business process modelling comprises two aspects – the control-flow perspective and data-flow perspective [1]. Control-flow perspective defines possible execution paths of a business process, while data-flow perspective represents how business objects are manipulated and change states during a process. Data in business process models are usually declared in terms of business objects (physical or virtual) and usually there are prescribed allowed states of business objects contained in internal business policies, external legislative documents, standards, reference models, and other regulations. Nowadays there is an increased pressure on organizations to guarantee compliance of their business processes with various regulatory and legislative requirements, other externally imposed constraints, and internal business policies [2]. For an organization engaged in business process modelling this might mean that (1) activities in business process models have to be associated with business objects representing inputs or outputs, (2) it has to be possible to represent a state of a business object at a given point of time, (3) it has to be possible to associate allowed state transitions with a business process model, and (4) it has to be possible to detect if a state of a business object is compliant with allowed state transitions. In this paper we are not talking about the soundness of the process – correctness criteria that a process model has to fulfil, e.g., deadlock or livelock patterns.

Compliance can be checked during or after the execution of the business process, called *compliance by detection*, or compliance can be checked while modelling the business process, called *compliance by design* [3]. In this paper we address the issue of compliance between business process models and lawful state space of business objects. In our solution we intend to apply *compliance by detection* method to check during the execution of the business process if states of business objects are compliant with the lawful state space. However, we also intend to generate a space of conceivable states for business objects at a design-time of business process.

We motivate our research with the following: compliance between business process models and lawful state space of business objects (1) ensures that organization will not violate laws and there will be no potential legal problems for the organization, and (2) ensures consistency in collaborative business processes and customer satisfaction. A number of studies exist that show the importance of addressing data and states of data in business process models – e.g., in [4] authors indicate the importance of *data-driven* process structures in large engineering processes such as assembling of a car or an airplane, and according to [5] in order to achieve safe execution of a process model it must be ensured that every time a task attempts to access a data object, the data object is in a certain expected data state (legal state). And since not all possible transitions of states are meaningful, restrictions on object state transitions are also required. In this paper we intentionally use the term "business objects" and not "data objects", since active structure elements are also capable of assuming a state which can be illegal and should be also monitored.

Nowadays organizations employ industry modelling standards like BPMN [6] and ArchiMate [7] to understand and improve business processes. Business Process Model and Notation (BPMN) [6] is the de-facto standard for representing in a very expressive graphical way the processes occurring in virtually every kind of organizations [8]. However, BPMN has its limitations when it comes to modelling other aspects of organizations such as organizational structure and roles, functional breakdowns, data, strategy, business rules and technical systems [9]. Information about Enterprise Architecture (EA) is needed to create real-world business process models. To provide a uniform representation for diagrams that describe EA, ArchiMate modelling language has been developed [7]. The core of ArchiMate language consists of three main types of elements: active structure elements, behaviour elements, and passive structure elements (objects) [7]. Some tools like ARIS [10] and QPR [11] allow linking BPMN and ArchiMate models in their modelling environments. Linkage between BPMN models and ArchiMate models provides possibilities to complement BPMN models with enterprise aspects and ArchiMate models with detailed process descriptions. In this paper we particularly address linked BPMN and ArchiMate models, which we, for simplicity reasons, call *business process models*.

The previous research has shown (see [12], [13] and [14]) that BPMN and ArchiMate lack in ability to describe flow of business objects in business process models and explicitly declare states of business objects imposed by regulations. This gap hinders compliance of business process models with external and internal regulations.

Wand and Weber [15] built a set of models for the evaluation of modelling techniques based on an upper ontology defined by Bunge [16]. They extended Bunge's ontology and applied it to the modelling of information systems (BWW model) [15]. BWW model consists of constructs present in the real world that must be represented in information systems. BWW model allows straightforwardly addressing: (1) states of things, (2) lawful state space and lawful event space of things, (3) conceivable state space and conceivable event space of things, (4) state law that restricts values of the properties of things to a lawful subset, and (5) lawful transformations that define which events in things are lawful. To be able to control whether an unlawful event has occurred in a business process, or a business object has assumed an unlawful state, it is necessary: (1) to provide means explicitly defining states of business objects in business process models (2) to generate lawful and conceivable states spaces for business process models, and (3) to check compliance of business process models with generated lawful state spaces at a run-time.

This paper presents an on-going research which aims to provide a solution and a prototype of a tool for supporting explicit declaration of lawful states and compliance checking between business process models and lawful state space of business objects. For a theoretical foundation purpose we propose to use BWW model [15], since BWW model complements BPMN and ArchiMate for what they are lacking – explicit representation of business objects, their states, and state transition laws.

Research presented in [17] describes how BPMN and ArchiMate support BWW model. There are 6 BWW model elements that are not supported by these modelling languages, namely, *State Law (SL), Conceivable State Space (CSS), Lawful State Space (LSS), History (H), Conceivable Event Space (CES),* and *Lawful Event Space (LES)*: or a tuple {SL, CSS, LSS, H, CES, LES}. These six elements are to be taken into consideration to define a complete, lawful, and consistent description of business processes. Our work focuses on the use of BWW elements {SL, CSS, LSS, H, CES, LES} in designing compliant with the states of business objects business process models. However, we are aware that the subject of compliance is broader than concerns of business object states.

The main contribution of this paper resides in that we use BWW model – a system's model with a proven research record – to supplement BPMN and ArchiMate models with explicit declarations of object states, state laws and conceivable and lawful state spaces in order to support organizations in achieving compliance with regulations.

The paper is structured as follows. In Section 2 the related work is outlined. In Section 3 a running example that we use throughout the paper is described. Section 4 contains formalization of BWW elements {SL, CSS, LSS, H, CES, LES} using a set theory. In Section 5 existing gaps and the proposed solution is discussed. Brief conclusions and future work are presented in Section 6.

2 Related Works

The lack of consistent theoretical foundation for building information systems urged Wand and Weber [15] to build a set of models for the evaluation of modelling techniques. Wand and Weber have extended the ontology presented by Mario Bunge [16] and developed a formal foundation called BWW model for modelling information systems [15]. Elements in BWW model (in the text shown in italics) can be organized in the following groups (adapted from [17]):

1. *Thing* – including *Properties*, *Classes* and *Kinds* of *Things*. *Thing* is an elementary unit in BWW. *Things* possess *Properties*, which defines *States* of a *Thing*. *Things* can belong to *Classes* or *Kinds* depending on a number of common *Properties*. A *Thing* can act on another *Thing* if its existence affects the *History* of the other *Thing*. *Things* are coupled if one *Thing* acts on another.
2. *State of Thing* – *Properties* of *Things* define their *States*. *State Law* restricts *Values* of *Properties* of *Things*. *Conceivable State Space* is a set of all *States* a *Thing* can assume. *Lawful State Space* defines *States* that comply with *State Law*. *Stable State* is a *State* in which *Thing* or a *System* will remain unless forced to change by *Thing* in the *System Environment*. *Unstable State* is *State* that will be changed into another *State* by the *Transformations* in the *System*. *History* is the chronologically-ordered *States* of *Thing*.
3. *Transformation* – transformation between *States of Things*. *Transformation* is a mapping from one *State* to another. *Lawful Transformation* defines which *Events* in *Thing* are lawful.
4. *Event* – is a change in *State* of *Thing*. *Conceivable Event Space* is a set of all *Events* that can occur to *Thing*. *Lawful Event Space* is a set of all *Events* that are lawful to *Thing*. *Events* can be *Internal Events* and *External Events*. *Events* can be *Well-Defined – Event* in which the subsequent *State* can be predicted – or *Poorly-Defined Event* in which the subsequent *State* cannot be predicted.
5. *System* – a set of coupled *Things*. *System Composition* is *Things* in the *System*. *System Environment* is *Things* outside the *System* interacting with the *System*. *System Structure* is a set of couplings that exist among *Things*. *Subsystem* is *System* whose composition and structure is a subset of the composition and structure of another *System*. *System Decomposition* is a set of *Subsystems*. *Level Structure* is an alignment of the subsystems.

The authors of [5] propose a notion of "weak conformance" which checks conformance of a process model with respect to data objects. This notion can be used to tell whether in every execution of a process model each time a task needs to access a data object in a particular state, it is ensured that the data object is in the expected state or can reach the expected state and, hence, the process model can achieve its goals. In [18] authors identify that consistency between business process models and object life cycle is required, however, their relation is not well understood. Authors clarify this relation and propose an approach to establish the required consistency by explicitly defining object states in business process models and then generating life cycles for each object type in the process. The authors of [18] indicate that object life cycle modelling is valuable at the business level. However, we propose to consider states of objects also at the application and technology levels of enterprise architecture since objects can be hidden and specified in sub-process structures at different levels of an enterprise. The authors of [19] use object life cycle as a common means for explicitly modelling allowed state transitions of an object during its existence and propose a technique for generating a compliant business process model from a set of given reference object life cycles.

The notion of a "legal state" is also mentioned in [20] where authors indicate that the representation of legal states in a model of a trade procedure is essential because organizations should be able to derive their obligations, rights, and duties at each point during the execution of the trade procedure and propose to annotate the states in Petri nets. In [2] authors investigate the use of temporal deontic assignments on activities as a means to declaratively capture the control-flow semantics that reside in business regulations and business policies. In object-oriented paradigm, state machines are extensively used for representation of states of objects [21]. In [22] the authors propose logic based formalism for describing the semantics of business contracts and the semantics of compliance checking procedures and close the gap between business processes and business contracts. In [3] the author focuses on compliance by design and extends artifact-centric approach to model compliance rules using Petri nets and show how compliant business processes can be synthesized automatically from the point of view of the involved business objects.

Since we address the importance of explicitly representing business objects and their states in business process models, our approach is also related to case handling [23] – a relatively new paradigm that, unlike workflow management, is strongly based on data. In our approach we generate a lawful state space using a conceivable state space based on a particular business process scenario (case).

The objective of this paper differs from the related work in that it uses BWW model as a theoretical foundation for generating conceivable and lawful state spaces from a business process model and applies it to nowadays de-facto modelling methods BPMN and ArchiMate.

3 Example: Electronic Submission

Throughout this paper we are using a simple electronic submission example at a university in which a researcher uploads his publication to university repository and can choose an option to publish her work as Open Access publication (see Figure 1). Researchers must choose a licence under which they wish to publish their publication – a version of the full text of the work which the publisher permits to archive in the institutional repository. The possible versions of the publication's full texts are: pre-print, post-print or published version. Uploaded publication can assume several states based on the set of its properties, e.g., lawful state will be when a version of a publication's full text is the pre-print and publisher has permitted archiving this publication. Lawful event will be allowing showing a full text of this publication publicly. Unlawful event will be when a publisher has not allowed archiving but a full text is made available publicly.

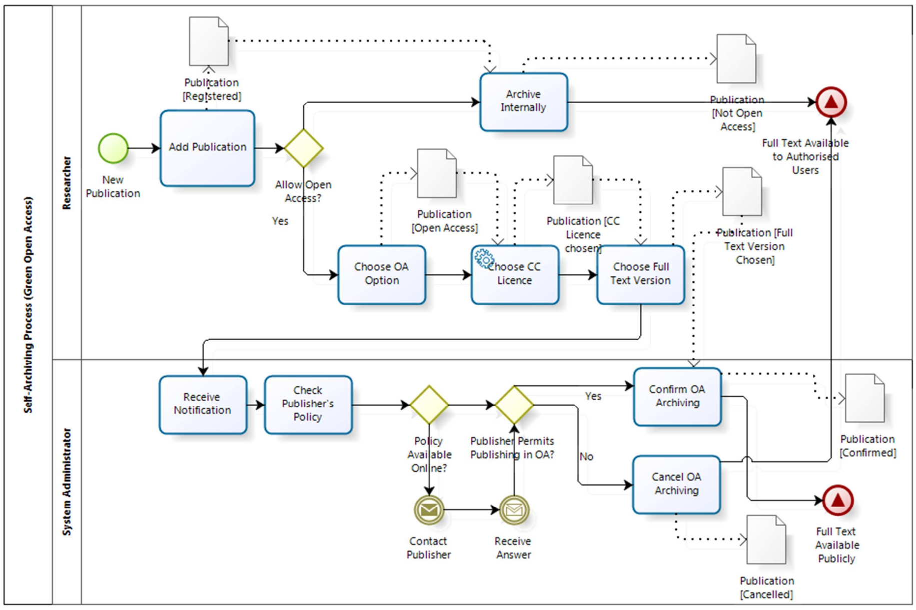


Fig. 1. BPMN 2.0 model of the electronic submission business process.

4 Formalization of BWW Model

In this section we propose formal definitions of BWW model elements based on informal description of BWW model presented in [12].

<u>Definition 1: *Thing*</u>. *A Thing is the elementary unit in the BWW ontological model. The real world is made up of Things. Things possess Properties.*
A *Thing* is a tuple:

T = {P, SL, CSS, LSS, H, LT, CES, LES}, where:

- P is a set of *Properties* of a *Thing*
- SL is a *State Law* of a *Thing*
- CSS is a *Conceivable State Space* of a *Thing*
- LSS is a *Lawful State Space* of a *Thing*
- H is a *History* of a *Thing*
- LT is *Lawful Transformation* of a *Thing*
- CES is a *Conceivable Event Space* of a *Thing*
- LES is a *Lawful Event Space* of a *Thing*

Example. In the running example presented in Section 3 *Thing* is a Publication submitted by a Researcher.

<u>Definition 2: *Property*</u>. *A Property is modelled via a function that maps the Thing into some value.*
Property is a tuple:

P = {a, t}**,** where**:**

- a is an *Attribute* of a *Property*
- t is a *Property* type, namely, in general, in particular, hereditary, emergent, intrinsic.

Property is described as a function that maps a *Thing* from a set of *Properties* P_x to P_y:

$$(f : P_x \rightarrow P_y).$$

Example. In the running example presented in Section 3 Publication is assumed to have the following *Properties* (due to limitation of space we present only a subset of all possible properties):

- P_1={Title, In General}.
- P_2={Status, In General} – differs from the notion *State* (although names can be identical). Values of "Status" can be "Registered, "Confirmed", "Cancelled".
- P_3={Open Access Mark, In General} – represents whether a Researcher has chosen the option to archive Publication as Open Access.
- P_4={CC Licence, In General} – represent chosen CC License, possible values: "CC BY", "CC BY-SA", "CC BY-ND".
- P_5={Version of the Full Text, In General} – can have values "pre-print" "post-print", or "publisher's version/PDF".
- P_6={Publisher Policy, In General} – can have values "Green" (can archive pre-print and post-print or publisher's version/PDF), "Yellow" (can archive pre-print), "White" (archiving not formally supported).

Definition 3: *State*. *The vector of values for all Property functions of a Thing.* Let's assume that there is Publication X, then a *State* for a Publication X at a given point of time can be defined as

$$S_{xi} = \{ID, \{P_1, P_2, ...P_i, P_{i+1}, ...P_n\}\}, \text{ where:}$$

- ID is a name that identifies the *State*
- $\{P_1, P_2, ...P_i, P_{i+1}, ...P_n\}$ is the vector of values for all *Property* functions

Example. *State* for a Publication X from the running example:

S_{Px} = {Confirmed, {Title X, Confirmed, Yes, CC BY, Pre-Print, Yellow}}

Definition 4: *Conceivable State Space.* *The set of all States that the Thing might ever assume.*

CSS = {S, T}, where:

- S is a set of finite conceivable *States*
- T is a *Transformation* that is a mapping function, e.g., from *State* X to *State* Y: $(f_t: S_x \rightarrow S_y)$ – it is an association to a particular activity in the business process model.

Example. For any uploaded Publication X from our running example:

CSS_P = {{Registered, Add Publication}, {Open Access, Choose OA Option}, {Not Open Access, Archive Internally}, {CC Licence Chosen, Choose CC Licence}, {Full Text Version Chosen, Choose Full Text Version}, {Publication Confirmed, Confirm OA Archiving}, {Publication Cancelled, Cancel OA Archiving}}

Definition 5: *State Law.* *A State Law restricts the values of the Properties of a Thing to a subset that is deemed lawful.*

$$SL = \{P_{law}\}, \text{ where:}$$

P_{law} are *Properties* of a *Thing* that are lawful and is a subset of *Properties* of a *Thing*:

$$P_{law} \sqsubseteq P$$

Example. In the electronic submission example a *State* "Full Text Available Publicly" is lawful only in case when *Properties* of Publication are, .e.g.:

- P_1={Title="Title X", In General}
- P_2={Status= "Confirmed", In General}
- P_3={Open Access Mark= "Yes", In General}
- P_4={CC Licence="CC BY", In General}
- P_5={Version of the Full Text= "Pre-Print", In General}
- P_6={Publisher Policy = "Yellow", In General}

Definition 6: *Lawful State Space*. *The set of States of a Thing that comply with State Laws of the Thing.*

$$LSS = \{S, SL\} \text{ where:}$$

- S is a set of finite lawful *States*
- SL is a *State Law* – set of Properties that are lawful for a Thing in this particular state

Example. Let's assume that a Researcher has uploaded a particular Publication X, then:

LSS_{Px} = {{Registered, {Title X, Registered}}, {Open Access, {Title X, Registered, Yes}}, {CC Licence Chosen, {Title X, Registered, Yes, CC BY}}, {Full Text Version Chosen, {Title X, Registered, Yes, CC BY, Pre-Print, Yellow}}, {Publication Confirmed, {Title X, Confirmed, Yes, CC BY, Pre-Print, Yellow }}

Definition 7: *History*. *The chronologically ordered states that a Thing traverses in time.*

$$H = \{s_s, s_i, \ldots, s_n, \ldots s_e\}, \text{ where:}$$

- s_s is a start *State*
- s_i and s_n are chronologically next *States* in time
- s_e is an end *State*

Example. *History* of *States* in the running example for a Publication X:

H_{Px} = {Registered, Open Access, CC Licence Chosen, Full Text Version Chosen, Confirmed}

Definition 8: *Lawful Transformation*. *Defines which Events in a Thing are lawful. Event is a change in a State of a Thing.*

$$LT = \{E_l, SC, CA\}, \text{ where:}$$

- E_l is a set of *Events* that are lawful in a *Thing*, it can be defined as a subset of all *Events:* $E_l \sqsubseteq E$
- SC is a set of *Stability Conditions* that specify the *States* that are lawful under *Lawful Transformation*
- CA is a set of *Corrective Actions* that specify how the values of the *Property* functions must change to provide a *State* acceptable under transformation law.

$$CA=\{(f:Px \rightarrow Py)\}$$

Example. In the running example LT for a Publication X in a *State* "Registered":

LT_{Px}={{E_1={Registered→Open Access}, SC_{E1} ={Registered, Open Access}, CA $_{E1}$ ={{Title X, Registered}→{Title X, Registered, Yes}}}, {E_2={Registered→Not Open Access}, SC_{E1} ={Registered, Not Open Access}, CA $_{E1}$ ={{Title X, Registered}→{Title X, Registered, No}}}}

<u>Definition 9: *Conceivable Event Space*</u>. *The set of all possible Events that can occur in the Thing.*

CES = {E, T}, where:

- E is a set of all Events that can occur in a *Thing*
- T is a *Transformation* that is a mapping function, e.g., from *State* X to *State* Y: $(f_t: S_x \rightarrow S_y)$

Example. In the running example CES for Publication X:

CES_{Px} ={{E_1{Registered→Open Access}, E_2{Registered→Not Open Access}, E_3{Open Access→CC Licence Chosen}, E_4{CC Licence Chosen→Full Text Version Chosen}, E_5{Full Text Version Chosen→Publication Confirmed}, E_6{Full Text Version Chosen→Publication Cancelled}

<u>Definition 10: *Lawful Event Space*</u>. *The set of all Events in a Thing that are lawful.*

LES = {E_l, LT} where:

- E is a set of lawful *Events*
- LT is a *Lawful Transformation*

Example. In the running example LES for Publication X:

LES_{Px}={E_1{Registered→Open Access}, E_2{Open Access→CC Licence Chosen}, E_3{CC Licence Chosen→Full Text Version Chosen}, E_4{Full Text Version Chosen→Publication Confirmed}}.

The applications of above-presented formalizations will be shown in Section 5.

5 Existing Gaps and Proposed Solution

This paper continues the research presented in [14] and [13] where the evaluation of BPMN and ArchiMate against BWW was presented. Based on the results presented in previous works, we can conclude that BWW model defines a set of elements that are supported by BPMN and ArchiMate modelling language as well as a set of elements that are not supported by these modelling languages. Majority of BPMN and ArchiMate core elements can be mapped to BWW constructs. However, it is necessary to supplement BPMN and ArchiMate modelling languages with the missing elements in order to be able to maintain a set of lawful object states in business process models.

Because in BPMN and ArchiMate there is no explicit representation for object's *State*, *Conceivable State Space*, *Lawful State Space*, *State Law*, *Conceivable Event Space*, *Lawful Event Space*, and *History* – resulting BPMN and ArchiMate models may be irrelevant and modellers may need to incorporate additional modelling techniques to overcome these defects. It may be impossible to detect from BPMN and ArchiMate models which events and states should be expected to occur and which events and states can occur but are illegal. Another important aspect is lacking of element *History*, which chronologically describes state changes of business objects. This deficiency can lead to problems regarding maintaining system's log and recovery.

These gaps hinder lawfulness of business process models, because lawful states of business objects are not explicitly depicted in business process models, models might contain meaningless states and events, since a set of conceivable states and events are not depicted, and, as a result, business process models do not represent real-world processes and can lead to business process incompliance with regulations. Also, since BPMN proclaims to be directly executable, omitting states and state transition laws may hinder correct automated execution.

Using BWW model will potentially support creating business process models compliant with regulations, since missing BWW elements are addressed. Our approach intends to achieve the following:

- Explicitly defining *Properties* of business objects in business process models using formal definition described in Section 4 and indicating whether business object is an input or output parameter of an activity.
- Explicitly defining *States* of business objects in business process models using formal definition described in Section 4.
- At business process design-time we intend to generate automatically *State Law*, *Conceivable State Space* and *Conceivable Event Space* directed graphs based on formal definitions presented in Section 4 and explicitly defined *Properties* and *States* of business objects.
- We intend to check compliance of business process with lawful states of business objects at a run-time. At business process run-time based on a particular process scenario or case, we intend to generate automatically *Lawful State Space*, *History*, and *Lawful Event Space* directed graphs.
- We intend to use rules for object life cycle generation presented in [18] for automatically generating conceivable and lawful state spaces. Rules for object life cycle generation presented in [18] are based on patterns that are matched in the business process model and used to create object life cycle with state transitions from initial state to possible end states.

The proposed solution for maintaining lawful states of business objects in business process models requires a repository-based modelling tool that accommodates BPMN, ArchiMate and BWW.

For the running example of electronic submission of a research paper to a university repository Figure 2 depicts *Conceivable State Space* and *Lawful State Space* graphs for a Publication X. We would like to indicate that Publication is not the only business object in this example – also "Notification from Publisher" is a business object, CC licence, etc., but due to limited space we do not add analysis of other business objects. *Conceivable State Space* and *Lawful State Space* graphs were created using formalisms defined in Section 4:

1. LSS was created using formal definition LSS = {S, SL} – which represents a sequence of *Lawful States* and what are *Properties* of *Thing* for the lawful states:

LSS_{Px} *= {{Registered, {Title X, Registered}}, {Open Access, {Title X, Registered, Yes}}, {CC Licence Chosen, {Title X, Registered, Yes, CC BY}}, {Full Text Version Chosen, {Title X, Registered, Yes, CC BY, Pre-Print, Yellow}}, {Publication Confirmed, Title X, Confirmed, Yes, CC BY, Pre-Print, Yellow}}.*

2. CSS was created using formal definition CSS = {S, T} – which represents all possible sequences of states for Publication:
CSS_P = {{Registered, Add Publication}, {Open Access, Choose OA Option}, {Not Open Access, Archive Internally}, {CC Licence Chosen, Choose CC Licence}, {Full Text Version Chosen, Choose Full Text Version}, {Publication Confirmed, Confirm OA Archiving}, {Publication Cancelled, Cancel OA Archiving}}.

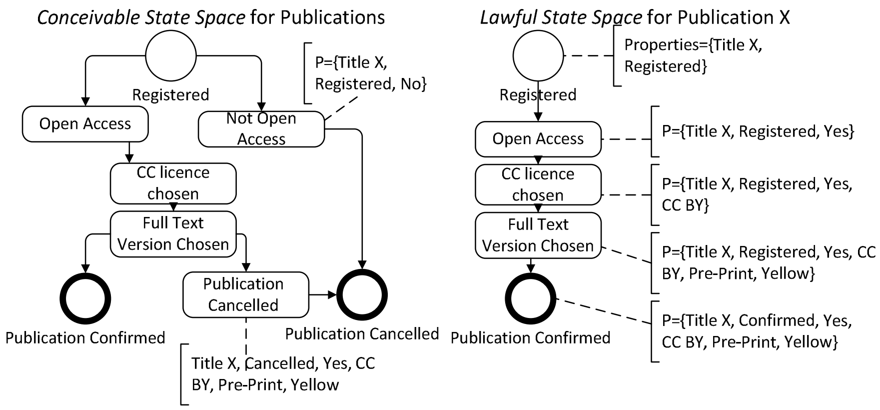


Fig. 2. Conceivable and lawful state spaces for a publication in electronic submission example.

6 Conclusions

Compliance between business process models and object state spaces are especially required in data-driven processes – in any process model that is based on data and manipulates with business objects. This paper presents an on-going research towards supporting compliance between business process models and lawful state space of business objects. BWW model is used as the foundation, since it allows straightforwardly addressing the lawful and conceivable state spaces of business objects. BPMN and ArchiMate modelling languages do not have elements that support explicit declaration of object states, including *State Law*, *Conceivable State Space*, *Lawful State Space*, *History*, *Conceivable Event Space*, and *Lawful Event Space*. The main contribution of this paper is a formalized solution for providing compliance between business process models and lawful states of business objects that has a capacity to support organizations in ensuring compliance between business process models and regulations.

With regards to tool support further research involves implementation of modelling environment capable of maintaining state spaces of business objects.

References

1. Weske, M.: Business Process Management. Springer Berlin Heidelberg, Berlin, Heidelberg (2012).
2. Goedertier, S., Vanthienen, J.: Designing compliant business processes with obligations and permissions. Bus. Process Manag. Work. 5–14 (2006).
3. Lohmann, N.: Compliance by design for artifact-centric business processes. Inf. Syst. 7241, (2013).
4. Müller, D., Reichert, M., Herbst, J.: A new paradigm for the enactment and dynamic adaptation of data-driven process structures. Adv. Inf. Syst. Eng. 48–63 (2008).
5. Meyer, A., Polyvyanyy, A., Weske, M.: Weak Conformance of Process Models with respect to Data Objects. ZEUS. 74–80 (2012).
6. OMG: Business Process Model and Notation 2.0, www.bpmn.org.
7. The Open Group: ArchiMate 2.0 Specification, http://goo.gl/7gC5B.
8. Chinosi, M., Trombetta, A.: BPMN: An introduction to the standard. Comput. Stand. Interfaces. 34, 124–134 (2012).
9. Silver, B.: BPMN Method and Style with Implementer's Guide. Cody-Cassidy Press (2011).
10. Softwareag: ARIS ArchiMate Modeler, http://goo.gl/WI76E.
11. Enterprise Architecture modeling software QPR EnterpriseArchitect, http://goo.gl/8Ots6h.
12. Rosemann, M., Recker, J., Indulska, M., Green, P.: A Study of the Evolution of the Representational Capabilities of Process Modeling Grammars. Adv. Inf. Syst. Eng. 447–461 (2006).
13. Penicina, L., Kirikova, M.: Towards Completeness and Lawfulness of Business Process Models. Perspect. Bus. Informatics Res. Lect. Notes Bus. Inf. Process. Vol.158, 63–77 (2013).
14. Penicina, L.: Linking BPMN, ArchiMate, and BWW: Perfect Match for Complete and Lawful Business Process Models? Short Pap. Proc. 6th IFIP WG 8.1 Work. Conf. Pract. Enterp. Model. (PoEM 2013). Vol-1023, 156–165 (2013).
15. Wand, Y., Weber, R.: On the ontological expressiveness of information systems analysis and design grammars. Inf. Syst. J. 3, 217–237 (1993).
16. Bunge, M.: Treatise on Basic Philosophy: Vol. 4: Ontology II: A World of Systems. (1979).
17. Recker, J., Indulska, M., Rosemann, M., Green, P.: Do Process Modelling Techniques Get Better ? A Comparative Ontological Analysis of BPMN. Campbell, Bruce, Underwood, Jim, Bunker, Deborah 16th Australas. Conf. Inf. Syst. (2005).
18. Ryndina, K., Küster, J., Gall, H.: Consistency of business process models and object life cycles. Model. Softw. Eng. 80–90 (2007).
19. Küster, J., Ryndina, K., Gall, H.: Generation of business process models for object life cycle compliance. Bus. Process Manag. 165–181 (2007).
20. Bons, R.W.H., Lee, R.M., Wagenaar, R.W., Wrigley, C.D.: Modelling inter-organizational trade using Documentary Petri Nets. Proceedings of the Twenty-Eighth Hawaii International Conference on System Sciences, vol.3. pp. 189–198. IEEE Comput. Soc. Press (1996).
21. OMG: UML 2.0 Superstructure specification, http://www.omg.org/spec/UML/2.0/Superstructure/PDF.
22. Governatori, G., Milosevic, Z., Sadiq, S.: Compliance checking between business processes and business contracts. 2006 10th IEEE International Enterprise Distributed Object Computing Conference (EDOC'06). pp. 221–232. IEEE (2006).
23. Aalst, W. Van der, Weske, M., Grünbauer, D.: Case handling: a new paradigm for business process support. Data Knowl. 1–36 (2005).

Ontology Matching and Alignment

A Method of Ontology-aided Expertise Matching for Facilitating Knowledge Exchange

Eduard Babkin, Nikolay Karpov, Tatiana Babkina

National Research University Higher School of Economics,
25/12 Bolshaja Pecherskaja Ulitsa, Nizhny Novgorod 603155, Russia,
{eababkin, nkarpov, tbabkina}@hse.ru

Abstract. The paper proposes a new method for facilitating knowledge exchange by seeking relevant university experts for commenting actual information events arising in the open environment of a modern economical cluster. This method is based on a new mathematical model of ontology concepts matching. We propose to use in the formal core of our method a new modification of Latent Dirichlet allocation. The method and the mathematical model of ontology matching were validated in the form of a software-based solution: the newly designed decision support system titled EXPERTIZE. The system regularly monitors different text sources in the Internet, performs document analysis and provides university employees with critical information about relevant events according a developed matching algorithm. In the proposed solution we made several contributions to the advances of knowledge processing, including: new modifications of topic modeling method suitable for application in expert finding tasks, integration of new algorithms and existing ontology services to show feasibility of the solution.

Keywords: expert finding, natural language processing, topic modeling.

1 Introduction

Emerging and successful growing of new forms of inter-organizational cooperation known as regional, innovation or university clusters [1] in national economies became a significant phenomenon of the modern world-wide socio-economical system. Sustainable exchange of expertise and professional knowledge between stakeholders of innovation clusters plays an important role in knowledge-based economics [2]. For this task an university undoubtedly should be a catalyst which provides expert evaluation and opinions. Critical problems and major strategic choices should be commented, discussed and exposed for multiple stakeholders including industry mass-media and society.

Until now there is no big success of tight integration of university community within the framework of emerging innovative clusters. Informational links are developed by *ad hoc* manner, major activities are implemented inside the stable university-based structures like incubators and business parks. Communication with business experts and mass media shows that in modern turbulent information environments it is the paradigm of information and knowledge exchange which should be modernized. The modernized paradigm of information and knowledge exchange should facilitate reactive or even proactive behavior of university community in response to critical emerging economic or social phenomena in the open environment of innovative cluster-based economy of knowledge.

Traditional analytical methods which provide modern university community with current information about important discussion topics and critical issues lack of comprehensiveness and become too slow. In nowadays practice of universities the best solutions primarily include manual analysis of mass media and internet resources and further slow distribution of information about relevant public events through the inefficient hierarchical organizational structure (from the schools, faculties towards department and employees).

We believe that advanced methods of automated and automatic knowledge management belong to critical scientific foundations of modernization the paradigm of information and knowledge exchange. A specifically designed combination of automated text processing and ontology-based knowledge engineering may improve quality of information analysis and reduce university's response time.

There are many interesting systems which approaches are close to our knowledge exchange idea. The one of it is Media Information Logistics project (Media-ILOG) which is concerns the domain of mass media too. The goal of the Media-ILOG [3], was to improve information flow inside a local newspaper JonkopingsPosten.

In our research we limited the scope of the aforementioned global problem to the key issue of real-time matching between relevant university experts and actual information events arising in the open environment of the economical innovation cluster. We offer a solution of that issue in the form of new automated method of experts finding for facilitating knowledge exchange between the university and heterogeneous community of the innovation cluster.

In contrast to Media-ILOG which is used semantic matching approach proposed by Billig et al. [4] The core of our method is a modification of Latent Dirichlet allocation. [5] It is algorithmically implemented in the newly designed decision support system titled EXPERTIZE. The system regularly monitors different text sources in the Internet, performs document analysis and provide university employees with critical information about relevant events according the specific relevance matching algorithm.

The high level design structure of EXPERTIZE software system includes several principal components. They are Crawler, Data Modeler, Data Store, GUI and Matcher. We match an input document not only with a single expert from our dataset, but with a scientific areas of interest, which is a category of the formal ontology. Each category is represented as a probability distribution of latent topics, so we match distribution of latent topics in the query document with the category using the maximum-likelihood estimation.

In the result of software implementation EXPERTIZE software system has been implemented as a software service. Now it is in an operating state, and regularly collects data from the several information resources available in Internet: library of HSE[9] and Elibrary[10]. Open systems interfaces allow EXPERTIZE get real-time access to the areas of domain interest of the employees of HSE from the InfoPort service [6].

[9] publications.hse.ru
[10] elibrary.ru

A set of practical use cases show that EXPERTIZE properly matches the actual information about discussion topics and information events.

The article has the following structure. After the introduction Section 2 contains related works overview in the information modeling and semantic matching to experts' domains. In Section 3 we observe essentials and formal foundations of our method. Main design decisions and functionality of EXPERTIZE software system are described in Section 4. Section 5 provides the readers with case study of application of that system in a real life information environment. Section 6 concludes the work, giving comparison results of our method and other known approaches and defining open research questions for further investigation.

2 Overview of Relevant Formal Methods for Expert Finding

As soon as our task is to match ontology concepts of expertise with plain text of news it is strongly related to the common expertise retrieval task. The past decade has appeared tremendous interest in expertise retrieval as an emerging subdiscipline. From 2005 the Enterprise Track at the Text REtrieval Conference (TREC) provided a common platform for researchers to empirically assess methods and techniques devised for expert finding [7]. The TREC Enterprise test collections are based on public facing web pages of large knowledge-intensive organizations, such as the World Wide Web Consortium (W3C) and the Commonwealth Scientific and Industrial Research Organisation (CSIRO).

Balog et al. 2012 [8] highlights state of the art models and algorithms relevant to this field. They classified expert finding approaches as follows:

- profile-based model;
- document-based model;
- hybrid model.

A profile-based model for expert finding using information retrieval proposed in Balog and de Rijke [9]. A candidate's skill is represented as a score over documents that are relevant given a knowledge area. The relevance of a document is estimated using standard generative language model techniques.

In the other approach, the method of document-based expert finding does not create a profile for each expert. It uses documents to match candidates to queries. The idea is to first find documents that are relevant to the topic and then locate the experts associated with these documents. The document models are also referred to as query-dependent approaches. Later, Fang and Zhai [10] presented a general probabilistic model for expert finding and showed how the document-based model can be adapted in this schema.

Balog et al. [8] applied this approach to a language model–based framework for expert finding. They also used the profilebased approach in their system and showed that the document-based approach performs better than the profile-based model. Serdyukov and Hiemestra [11] proposed a hybrid model for expert finding which combines both profile- and document-based approaches.

Semantic analysis of texts for expert finding with required competencies proposed by Fomichov on the basis of Formal Concept Analysis [12]. The approach allows to build and compare semantic representations of expert profile using the theory of K-representation and a model of linguistic database.

A topic modeling approach for expert finding proposed by Balog et al. [13]. Instead of modeling candidate profiles or documents, they built a model for each input query and used this model to calculate the probability of candidates given queries. Their approach is similar to the document likelihood method, which is used in language model–based information retrieval. Based on their results, this model underperforms the profile- and document based approaches. The main reason of its poor performance is the sparsity of the models built from the queries. Their definition of topic, however, is different from ours. The term topic in their work refers to query words that users use to search for experts, whereas in the present work we use the term topic as a set of concepts that are extracted from a collection using a topic modeling algorithm. There are multiple known methods for topic modeling of document which are Latent Semantic Analysis (LSA) [14] Latent Dirichlet allocation (LDA) [5] et al.

The topic modeling approach is based on the assumption that words in a document are independent of one another (bag of words) and of their order in the text. Similarly, documents in a corpus **D** are independent of one another and unordered. Distribution of words $\mathbf{W}$ is determined by the set of latent topics $\mathbf{Z}$. Each topic has its own word distribution (phi) and each document has distribution over topics (theta).

Traditional topic-based information retrieval approach is exploited by Wei and Croft, 2006 [15]. The extracted topics are used for information retrieval; whereas the to-be-retrieved documents are used in the retrieval step, i.e., the distribution of topics over words (phi) is used for estimating $P(\mathbf{Q}/\mathbf{Z})$, where $\mathbf{Q}$ – is a set of word in query. The distribution of documents over topics (theta) is used for estimating $P(\mathbf{Z}/\mathbf{D})$.

Another topic-based model is proposed by Momtazi and Naumann [16]. This model outperforms the state-of-the-art profile- and document-based models. To-be-retrieved documents are not used in the retrieval step. Instead, we only use these documents for training LDA, i.e., to be-retrieved documents are used as a corpus to extract topics in an off-line process. Then, in the retrieval step, we only use the distribution of topics over words (phi) for estimating both $P(\mathbf{Q}/\mathbf{Z})$ and $P(e/\mathbf{Z})$ where e – is an expert label.

In a paper [17] the researchers show how to use a topic-based model with scientific ontology, where each document labeled with a category in scientific classification taxonomy **C**. They represent each category *c* as a conditional probabilistic distribution $P(\mathbf{Z}/c)$ which denotes the probability of category *c* being labeled with topic *z*. By utilizing LDA, $P(\mathbf{Z}/c)$ is a $|\mathbf{Z}|$-dimension vector of topic distribution. The main requirement for this approach is to estimate the probability $P(z_k/c)$, which cannot be obtained directly from LDA. However, according to the Bayes formula authors calculate $P(z_k/c)$ by

$$P(z_k/c) = \frac{P(c/z_k)P(z_k)}{\sum_k P(c,z_k)} \tag{1}$$

where $P(c/z_k)$ and $P(z_k)$ can be obtained from LDA. As soon as $\sum_k P(c,z_k)$ is constant for different *c* and $P(z_k)$ is uniform distribution we have

$$P(\mathbf{Z}/c) \propto P(c/\mathbf{Z}) \qquad (2)$$

On the basis of explored papers the best way to solve our task is to match between relevant university experts and actual information events using topic-based model, which is proposed by Momtazi and Naumann [16]. Thus, we should implement the model for papers in Russian language concluded in our enterprise dataset. With the help of approach described in [17] this topic-based approach can be applied to use with scientific ontology. To show feasibility of the solution, we archive an integration of new algorithms and existing ontology services.

3 The Essentials and Formal Foundations of the Method Proposed

In our previously designed InfoPort system [6] for solving the expert finding problem we proposed to translate a user-specified query to a corresponding SPARQL query which is evaluated against a specific set of RDF repositories. The query result consisted of a relevant category of scientific classification taxonomies and keywords. The search algorithm of InfoPort system retrieved all persons who labeled with this query.

In the current research our new system EXPERTIZE works automatically: it gets news event as a query and matches it to the most relevant scientist, who can provide expert evaluation and opinions about it. In other word we arrange experts in order to relevance to the event.

On the one hand news events are represented as news in natural language format, thus we have ability to extract semantic information from the text. On the other hand each expert has texts in the form of written papers or records of spoken interviews and tutorials. This material contains rich semantic information about personal interests and abilities.

There are some formal models suitable for implementation of context analysis such as a Distributional Semantic Model (DSM) [18][19] and Latent Semantic Analysis [20][14] and Latent Dirichlet Allocation [5].

In our project we use an extension of Latent Dirichlet Allocation which is a generative formal model that uses latent groups to explain results of observations – data similarity in particular. For instance, if the observations are words in the documents, one can posit that each document is a combination of a small number of topics and that each word in the document is connected with one of the topics. Latent Dirichlet Allocation (LDA) is one of topic-modeling methods and was first introduced by its authors as a graphical model for topic detection.

In our approach by training the LDA model, we form the statistical portrait of its author. A person writing a text has a set of topics in their mind, and each document has a certain distribution of these topics. The author first selects the topic to write on; within this topic, there is a distribution of words that may occur in any document that contains this topic. The next word in the text is generated within the distribution. Then the same procedure is repeated. On each iteration, the author either selects a new topic or continues to use the previous one, and generates the next word within the active topic [5].

The first step of our method for expert finding is a training the model on a collection of texts. We get an estimate of two discrete distribution functions. The following is distribution of probabilities of words $\mathbf{W}$ in topics $\mathbf{Z}$:

$$P(w_i / z_k);\ i \in \overline{1,|\mathbf{W}|}, k \in \overline{1,|\mathbf{Z}|} \tag{3}$$

Distribution of probabilities of topics $\mathbf{Z}$ in documents $\mathbf{D}$:

$$P(z_k / d_n);\ k \in \overline{1,|\mathbf{Z}|}, n \in \overline{1,|\mathbf{D}|} \tag{4}$$

Semantic representation of query news document d_0 can be also calculated using built LDA model. It is distribution of probabilities of topics $\mathbf{Z}$ in documents d_0 – $P(z_k / d_0);\ k \in \overline{1,|K|}$.

In the second step, the extracted topics are used to calculate the probability of query document d_0 given candidates E and categories C. Both E and C represented as a word in the LDA model. Thus, P(Q/E) and P(Q/C) is calculated based on the topics that are distributed over candidate names (E) or scientific domain topics (C).

$$P(d_0 / E) = \sum_{z \in Z} P(d_0 / z, E) P(z / E) \tag{5}$$

$$P(d_0 / C) = \sum_{z \in Z} P(d_0 / z, C) P(z / C) \tag{6}$$

By assuming conditional independence between d_0 and E, C and the document d_0 as equiprobable with other documents we have

$$P(d_0 / z, E) = P(d_0 / z, C) = P(d_0 / z) = \frac{P(z / d_0) P(d_0)}{P(z)} \propto P(z / d_0) \tag{7}$$

Using (2) and (7) from (5) and (6) we get following simple formulas

$$P(d_0 / E) \propto \sum_{z \in Z} P(z / d_0) P(E / z) \tag{8}$$

$$P(d_0 / C) \propto \sum_{z \in Z} P(z / d_0) P(C / z) \tag{9}$$

We rank the categories C from scientific classification taxonomy according to the maximum-likelihood estimation. Most probable categories are chosen and associated with expert.

$$c_{\max} = \arg\max_{c \in C} \left(P(d_0 / c) \right) \tag{10}$$

We perform the same approach to rank experts E from a set of employees of the company.

4. Software Design of EXPERTIZE

The described method for experts finding was practically implemented during design and implementation of the system for matching between relevant university experts and actual information events arising in the open environment of the economical cluster. Such system was called EXPERTIZE. The following services are distinguished in the high-level design of that system (Fig.2):

1. Web Crawler;
2. Data Modeler;

3. Data Store;
4. Graphical User Interface (GUI);
5. Matcher.

EXPETIZE actively uses our InfoPort Service [2]. That semantic service provides in the form of formal ontology factual information about more than three hundred employees of Higher School of Economics (HSE NRU)[11] branch at Nizhny Novgorod. The InfoPort data is represented as RDF triples. Triples include hierarchical information as it originally is in the source. The first level is an alphabetical ordered list of group of scientist, second is a scientist with his personal interests and papers, and third is papers with its features.

The components of the EXPETIZE system can be classified as Online and Offline services. Both are interacted with InfoPort via native REST interface. Offline ones work within monthly period to update information regularly. Online services work on demand, when user activates it by web interface.

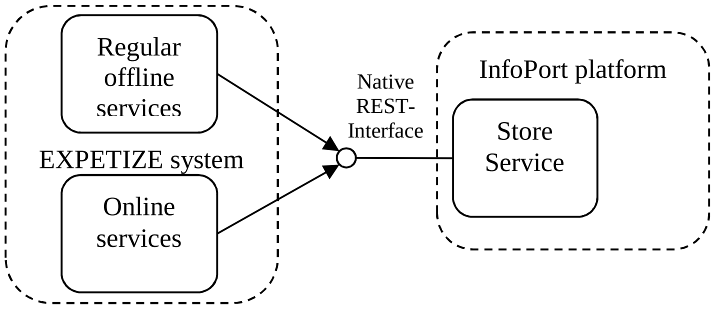


Fig. 1. Interaction of EXPETIZE services with InfoPort platform.

Offline processing begins with crawler Service by scheduler. It makes a request via REST-interface to the InfoPort Store Service to take a list of papers' URI (Uniform Resource Identifiers). As soon as each paper is available online the Crawler gets it by URI and extracts paper's features from page using XML parser. Paper's features include: authors, title, abstract, free keywords, scientific categories of ontology. This information is collected to the Data Store with the help of MySQL[12] base as a Temporal raw data. Implementation of Crawler Service uses Python[13] programming language and Lxml[14] library for HTML processing.

Preprocessing in the Data Modeler service includes the following steps:

- get temporal raw data;
- tokenize the text;
- lemmatize the tokens;
- index the words using the dictionary of lemmas;
- filter out the words that are too frequent (stop words) or too rare (used only once);
- index authors and scientific categories;
- form bag of words using lemmas, authors and categories;

[11] http://www.hse.ru/en/
[12] http://www.mysql.com
[13] http://www.python.org
[14] http://lxml.de

- build LDA model with a given number of topics *K*.

At present time, there are several methods for building LDA models, that is, methods of searching for parameters of all distribution functions in the model. All of the methods are iteration-based and are similar in structure to the Expectation Maximization (EM) algorithm. They are:

- Online Variational Bayes algorithm [21];
- Gibbs Sampling [22];
- Expectation Propagation [23].

Among these algorithms, we use the Online Variational Bayes algorithm as it is the most precise one [21]. It is well implemented in the Gensim[15] toolkit. resource-intensive algorithm.

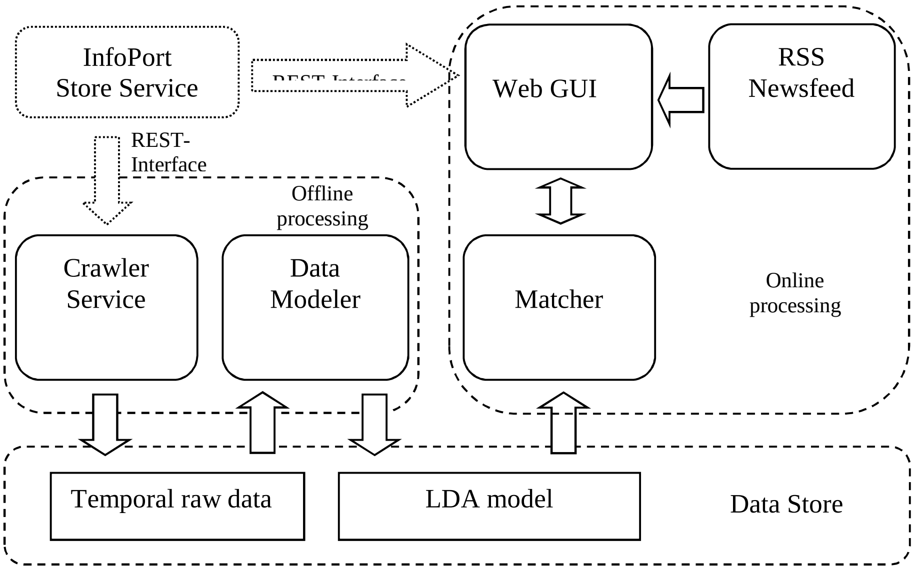


Fig. 2. Principle design of the EXPERTIZE system.

Online processing performs on demand of user by opening Web GUI. Web interface activates RSS Newsfeed, which gets and displays 10 last news from the RSS feed and an empty textbox. User can choose one of its 10 news or paste the text to the textbox manually. When user specify input query the GUI transfer it to the Matcher. In turn, this component performs online semantic search. A semantic representation of event is matched with semantic representations of scientific categories and experts by applying the formula (8) and (9) and selecting top 5 of the units. So, the Mather component returns 5 URIs to the GUI.

To provide user friendly output of the finding result GUI component makes a request to the Infoport Service. It gets features of the selected units: full name, expert's photo URL, expert's department.

[15] http://radimrehurek.com/gensim

The crawled collection includes 4132 units but only 1492 papers are in the Russian language. So, we decide to extend collection with the help of eLibrary[16] scientific database. This is a biggest scientific database in the Russian language. We extracted a part of this base connected only with Information Technologies field. It includes 9127 papers not older than 2011 year.

5 Case study

Evaluation of our proposed method and the EXPERTIZE system was performed empirically. We choose experts from our pool. This pool includes more than three hundreds of professors and researchers of the HSE NRU branch at Nizhny Novgorod[17]. According to an experts field of the study we chose news, which one can be comment by expert and put it to EXPERTIZE. If this expert appears in the list, proposed by the system, we mark such attempt as a successful match.

Let’s take a case study. There is an expert Sidorov Dmitry V. whose profile includes a set of scientific domain topics, which he is interested in. There are:

- w_1 - innovation projects,
- w_2 - venture investments,
- w_3 - innovative potential estimation
- etc.

Each scientific domain topics coded as one word and we have pre-created table which is distribution of probabilities of words $\mathbf{W}$ in latent topics $\mathbf{Z}$: $P(w_i / z_k);\ i \in \overline{1,|\mathbf{W}|}, k \in \overline{1,|\mathbf{Z}|}$. It usually has small number of elements higher than zero.

Table.1. Example of probabilities distribution of words $\mathbf{W}$ in latent topics $P(w_1 / z_k); k \in \overline{1,|\mathbf{Z}|}$

	z_1	z_2	…	z_{58}	…	z_{200}
w_1	0	0.04		0.1		0

We find news with title «Yandex company pays for big data»[18], which he can be able to comment as an expert. This news is about investment of Russian IT giant to an Israeli startup company. As each other documents in the collection it can be found probabilities distribution of latent topics z in document. This news goes as an input to the Matcher component where it converts to the probability distribution over latent topics (4) using the pre-built LDA model. The number of topics we set equal to 200.

Table.2. Example of probabilities distribution of topics $\mathbf{Z}$ in documents d_0 – $P(z_k / d_0);\ k \in \overline{1,|K|}$.

	z_1	z_2	…	z_{58}	…	z_{200}
d_0	0	0.21		0.058		0.034

[16] http://elibrary.ru
[17] http://nnov.hse.ru/en/
[18] http://www.kommersant.ru/doc/2469831

Next, using formula (10) the algorithm chooses each categories ***c*** from scientific classification taxonomy and finds $P(d_0 / C)$. Top 5 of experts who has maximum $P(d_0 / C)$ is shown in the system. A result is presented in Fig. 3.

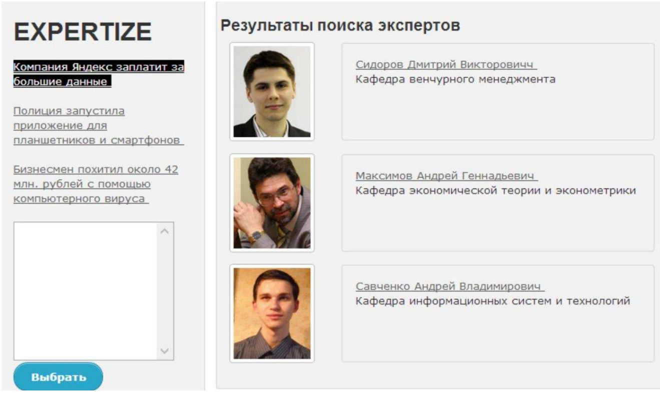


Fig. 3. Graphical user interface of the EXPERTIZE system.

As soon as our target expert Sidorov Dmitry V. is presented in the output we mark this trial as a successful one. From 100 trials we get 43 successful matches.

Table 3. Experimental results with different model of ontology matching.

Document-based	0.31
Candidate-based	0.22
Topic-based	0.43

We choose the Topic-base approach because in comparison with other approaches (Document-based and Candidate-based) this one gets the best results.

6 Conclusion

In this article we presented a new approach to support rapid exchange of knowledge in innovation clusters based on reactive experts finding. The proposed method of expert finding uses open Internet resources and existing ontological services like InfoPort [6] to get access to the approved skills of potential experts.

During our research we developed a new formal method based on Latent Dirichlet allocation, which includes a software-based solution for matching between relevant university experts and actual information events arising in the open environment of the economical cluster. This solution allows performing real-time matching between Internet news and areas of interest of university employees with further quick notification about possible participation of relevant employees in interviews, informational programs and discussions. In the proposed solution we made several contributions to the advances of knowledge processing, including: new modifications of topic modeling method suitable for application in expert finding tasks, integration of new algorithms and existing ontology services to show feasibility of the solution**.**

A software design of decision support system EXPERTIZE was developed for practical application of the method proposed. The first use cases of the EXPERTIZE system show their relevance and ability to solve the task specified.

Using topic-based model proposed by Momtazi and Naumann [16] we have achieved about 0.43 amount of mean average precision (MAP) on our own queries. The same approach on TREC 2005 and 2006 queries, shows 0.248 and 0.471 amount of MAP respectively [16]. So, precision of EXPERTIZE system is not much less than achieved on TREC 2006. The estimation of recall and f-measure in our EXPERTIZE system less interesting because in general user doesn't need a full set of various experts. One or two most relative experts usually enough for facilitating knowledge exchange.

As soon as we perform expert matching with scientific categories we can apply cross-language expertise retrieval by applying multi-language scientific ontology. It would be our prospective work.

References

1. Asheim, B., Cooke, P., Martin, R.: Clusters and regional development: Critical reflections and explorations. Econ. Geogr. 84, 109–112 (2008).
2. Klimova, N., Litvintseva, M.: Universities Innovation Clusters: Approaches for National Competitiveness Paradigm, (2011).
3. Sandkuhl, K., Öhgren, A., Smirnov, A., Shilov, N., Kashevnik, A.: Ontology construction in practice: Experiences and recommendations from industrial cases. 9th International Conference on Enterprise Information Systems, 12-16, June 2007, Funchal, Madeira–Portugal (2007).
4. Billig, A., Blomqvist, E., Lin, F.: Semantic matching based on enterprise ontologies. On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS. pp. 1161–1168. Springer (2007).
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation, (2003).
6. Babkin, E., Karpov, N., Kozyrev, O.: Towards Creating an Evolvable Semantic Platform for Formation of Research Teams. In: Kobyliński, A. and Sobczak, A. (eds.) Perspectives in Business Informatics Research. pp. 200–213. Springer Berlin Heidelberg (2013).
7. Craswell, N., de Vries, A.P., Soboroff, I.: Overview of the TREC 2005 Enterprise Track. Trec. pp. 199–205 (2005).
8. Balog, K., Fang, Y., de Rijke, M., Serdyukov, P., Si, L., others: Expertise Retrieval, (2012).
9. Balog, K., De Rijke, M.: Determining Expert Profiles (With an Application to Expert Finding). IJCAI. pp. 2657–2662 (2007).
10. Fang, H., Zhai, C.: Probabilistic models for expert finding. Advances in Information Retrieval. pp. 418–430. Springer (2007).
11. Serdyukov, P., Hiemstra, D.: Modeling documents as mixtures of persons for expert finding. Advances in Information Retrieval. pp. 309–320. Springer (2008).

12. Fomichov, V.: Semantics-Oriented Natural Language Processing: Mathematical Models and Algorithms. Springer (2009).
13. Balog, K., Bogers, T., Azzopardi, L., De Rijke, M., Van Den Bosch, A.: Broad expertise retrieval in sparse data environments. Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 551–558. ACM (2007).
14. Dumais, S.T.: Latent semantic analysis. Annu. Rev. Inf. Sci. Technol. 38, 188–230 (2004).
15. Wei, X., Croft, W.B.: LDA-based document models for ad-hoc retrieval. Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 178–185. ACM (2006).
16. Momtazi, S., Naumann, F.: Topic modeling for expert finding using latent Dirichlet allocation, (2013).
17. Zhu, H., Cao, H., Xiong, H., Chen, E., Tian, J.: Towards expert finding by leveraging relevant categories in authority ranking. Proceedings of the 20th ACM international conference on Information and knowledge management. pp. 2221–2224. ACM (2011).
18. Baroni, M., Lenci, A.: Distributional memory: A general framework for corpus-based semantics, (2010).
19. Turney, P.D.: Similarity of semantic relations, (2006).
20. Landauer, T.K., Foltz, P.W., Laham, D.: An introduction to latent semantic analysis. Discourse Process. 25, 259–284 (1998).
21. David M. Blei, Matthew D. Hoffman: Online Learning for Latent Dirichlet Allocation, http://books.nips.cc/papers/files/nips23/NIPS2010_1291.pdf, (2010).
22. Thomas L. Griffiths, Mark Steyvers: Finding scientific topics, (2004).
23. Thomas Minka, John Lafferty: Expectation-propagation for the generative aspect model. Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence. pp. 352–359. Morgan Kaufmann Publishers Inc. (2002).

Enhancing Alignment Results in Ontology Matching for Smart Cities

Lorena Otero-Cerdeira, Francisco J. Rodríguez-Martínez and Alma Gómez-Rodríguez

LIA2 Research Group. Computer Science Department. University of Vigo. Spain
{locerdeira, franjrm, alma}@uvigo.es

Abstract. In this paper we propose the use of an *ontology matching* algorithm to guarantee the interoperability of the different agents that integrate an smart city. In this sort of environment the different parties need to cooperate and to integrate their information in order to provide enhanced services to the users of the smart city. As the information of these parties may be described by means of different and heterogeneous ontologies, we find the solution in the use of ontology matching techniques. The algorithm presented was designed to be able to exploit the knowledge of previous matched agents to enhance the results and provide the most accurate results possible.

Keywords: internet of things, smart cities, ontology, ontology matching, alignment reuse

1 Introduction

In the last years there has been a remarkable increase in the amount of projects and initiatives related to *Internet of Things* [1] and *Smart Cities* [2]. The Internet of Things is the evolution of the information and communication technologies (ICT), that is taking us from having connectivity at anytime and anyplace to also having it with anything. This situation is reflected by the growing amount of different devices with connecting capabilities, such as RFID tags, NFC devices, sensors, actuators, etc. Such devices are the building blocks of the smart cities.

The idea behind integrating these devices in a city is to turn it into a smart one, so citizen’s lives can be improved with new types of services and comfort. These services can be related to almost every aspect of city life and infrastructure, water and energy supply, transportation, healthcare, education, etc. [23], and precisely the cities are looking at ICT to offer services to citizens while reducing costs and improving efficiencies.

To turn a city into a smart one, the first task to address is to develop a rich environment of networks that support digital applications [2]. This task involves, firstly deploying the proper infrastructure which includes different types of sensors, smart devices and actuators, together with the actual networks that allow the communication of these. However, the devices by themselves are not enough and it is necessary to develop applications that exploit these networks of devices.

Hence, in a smart city, smart devices, Sensor Networks (SNs) [23] and applications to exploit them, assume a crucial role. These urban sensors and sensor networks are generally spread over a wide area and continuously measuring different variables. The data collected is processed by the different applications which may trigger an action in some actuator or the response to a user's request.

It is highly likely that the deployment of a smart city is not done all at once but in a series of steps, so it is equally likely that different parts of the smart city are developed by different parties, resulting in the coexistence of different public and private deployments, each one of which possibly using different smart devices and also their own hardware and software architectures. It is necessary to guarantee the success of a smart city to put a special interest in allowing that these different deployments will be able to interact and seamlessly communicate with each other and, that the information gathered by the different devices will be properly integrated and shared among the different systems [26].

This problem is not new to the research community and several alternatives have been already proposed [26][25][11]. These approaches propose using a wrap for the different sensors, or compel the use of some standard or protocol to allow the communication between parties with different knowledge representations. Other efforts include the use of ontologies to semantically describe services and devices available [3]. The work that we have developed is in line with the latest but what we propose exploits ontologies differently.

Our proposal includes the use of *ontology matching* techniques [4] to guarantee the connectivity among the different parties in a smart city.

The remaining of the paper is organized as follows. In section 2, we delve into the use of ontology matching in smart cities and provide the foundations that supported the development of our system. In section 3, a description of our solution is provided and discussed. Finally in section 4, the main conclusions and future lines of work are summarized.

2 Ontology Matching in Smart Cities

A smart city may be seen as a distributed system where several agents on behalf of their users collect data from the environment by using different sensors. The concept of user here should be globally understood, as the user of an agent may be a citizen, a smart device, an application, another agent, etc. The use of ontologies in smart cities is not new as there is for instance the *SCRIBE* ontology [24] [5] designed out of the information gathered from different cities or the $SOFIA^2$ platform [6]. Ontologies help in providing a vocabulary to describe a certain domain and the specification of the meaning of the terms in that vocabulary [4], in our concrete case, ontologies help in defining the different events, entities and services in a smart city. Besides, they are particularly suitable for describing the meaning of the concepts in a communication process between the agents in a Multi-agent System (MAS) [7] and hence they are used as a way of reducing the semantic gap among the different interacting parties.

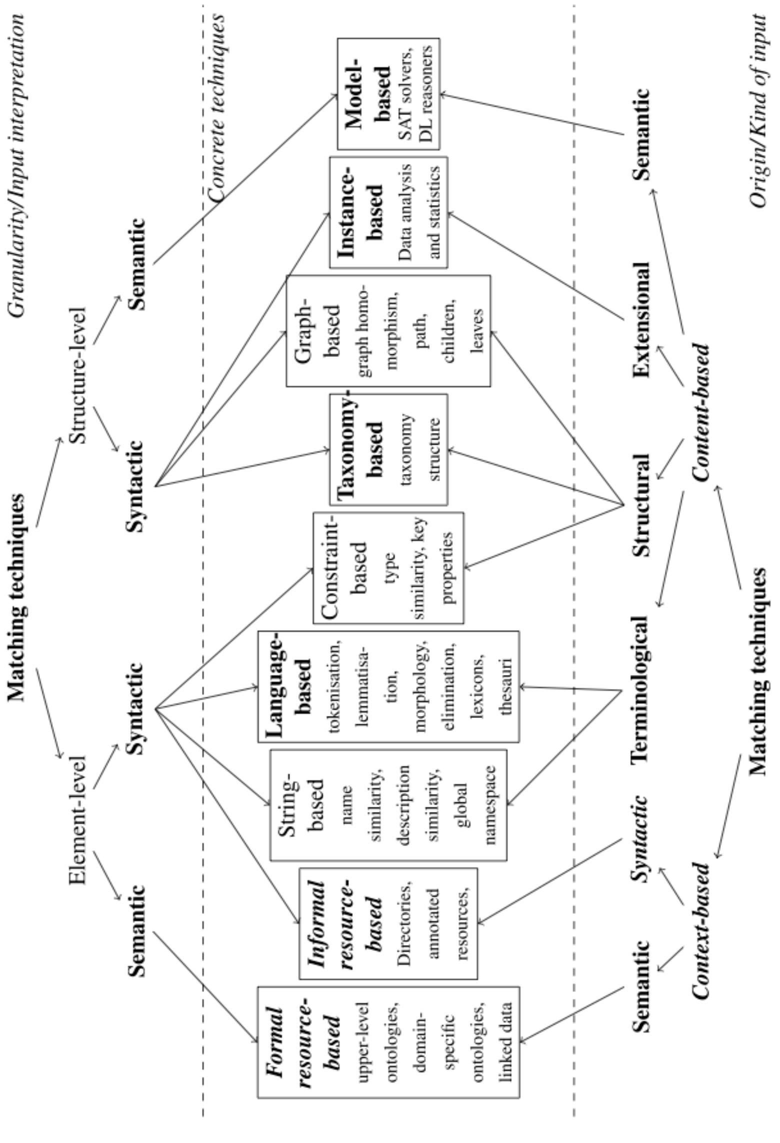


Fig. 1. **Fig. 2.** Classification of matching techniques

However, there are several reasons why ontologies by themselves are not enough to guarantee the interoperability of the different agents. For instance, the agents may use different ontologies to represent the information gathered from the sensors, the software applications in the smart city may be developed by different providers that represent their internal knowledge using different ontologies, there may be agents or applications included in the smart city in a later stage or even itinerant agents that only need a concrete service at a certain time. In order to actually reduce the heterogeneity in the definitions and allow a seamless communication of the parties, we relied on *ontology matching techniques* [8].

These techniques allow the identification of *alignments* for pairs of ontologies where an alignment identifies the set of correspondences holding between the entities belonging to the ontologies [9]. Apart from the manual identification of correspondences fulfilled by human experts which has been practically dismissed due to its cost, there are automatic and semi-automatic methods to compute the alignment between the ontologies which exploit different features of the ontologies or use external resources to identify the possible correspondences between the concepts.

Different classifications have been made for the matching techniques although for the scope of this paper, we followed the one that Euzenat and Shvaiko propose in [4]. This classification, as shown in figure 1[19], can be read both top-down, then stressing the interpretation that the different techniques provide for the input information, and also bottom-up, focussing on the type of input that the matching techniques use. Regardless of the direction of the reading, they both meet at the *concrete techniques* layer.

In the following section, while describing our solution to the ontology matching problem in smart cities, we briefly describe the different techniques that we have used linking them to this classification.

3 Solution Description

In this section we briefly describe our algorithm for ontology matching in smart cities and how we have enhanced its results by following an *alignment reuse* [12] approach.

It takes as input two OWL [18] ontologies and relies on the exploitation of some initial correspondences which we named *binding points* and which are similar to the *anchors* initially used by systems such as *LogMap* [13], *Anchor-Flood* [14], *Anchor-Prompt* [15] or *ASCO* [16], although the procedure followed to compute the binding points is remarkably different to the one used to obtain the anchors in each one of these systems.

These initial correspondences are obtained by using some *language-based* and *terminological techniques*. The language-based techniques consider names as words in a natural language and exploit their morphological features. Some of the methods used, as part of the pre-processing of the strings, are *tokenisation*, that consists of splitting words into shorter sequences by means of a separator (blanks, punctuation marks, camel-case changes, etc) and *stopword elimination*, that consists of removing words such as articles, prepositions, etc.

On the other hand, the terminological techniques consider their inputs just as strings and apply string-distance measures to asses the similarity between two entities. In our case we have used *Jaro-Winkler distance* [10] and *Levenshtein distance* [10] on the

[19] Extracted from the book Ontology Matching [4]

pre-processed strings. The results of these distances are weighted in order to obtain an only lexical value for each pair of entities in the ontologies to match. To weight the results of these measures, another similarity distance is used, in this case, it is based on the exploitation of *WordNet* [17] as a external resource. This is also a language-based technique that takes advantage of the definitions provided by this lexical database to evaluate the distance between two terms.

Once the similarity between the terms in the ontologies has been determined, only those pairs with the highest value are selected to become the *initial binding points*.

These initial correspondences sequentially undergo several procedures that take advantage of some *structural* features of the ontologies and that allow the discovery of new binding points. These binding points can identify both pairs of classes or properties. Each one of the newly discovered binding points is assigned a tag that identifies the procedure and branch within it that led to its discovery. If a binding point is reached by several procedures, all the tags are recorded.

1. *Properties Inverse Procedure:* this procedure retrieves new correspondences between properties by exploiting the existence of inverse properties defined with the construct *owl:inverseOf.*
2. *Properties Domain Range Procedure:* this procedure obtains new correspondences between classes by comparing the domains and ranges of the initial properties. Not only the first-level domain and range classes are evaluated but the procedure continues until reaching the higher levels of the hierarchy.
3. *Classes Properties Procedure:* this procedure allows the retrieval of both new correspondences between classes and properties. This procedure recursively identifies the similar properties existing among the class correspondences, and then assesses the existence of other classes belonging to the domain or range of this properties that could be a new correspondence.
4. *Classes Family Procedure:* this procedure retrieves new correspondences between classes. It exploits the familiar relations of the classes. For each pair of them, its superclasses, subclasses, and sibling classes are evaluated to determine the existence of new possible matches.

These procedures are iteratively applied until no new correspondences are discovered. Once these procedures have finished all the correspondences that have been discovered are filtered to produce the final output of the algorithm. To do so, the tagging is very important as it allows the identification of the different procedures and sub-procedures. It is based on the idea that the different procedures exploit different structural features of the ontologies and hence the likelihood that the obtained results are good is not the same for all of them.

To evaluate the performance of the algorithm we intended to use ontologies from the smart city domain. However, the amount of ontologies in this area proved not to be enough to allow an accurate evaluation. Hence we have used the testbed provided by the *Ontology Alignment Evaluation Initiative 2013* [19] (OAEI-13) which provides different series of tests to evaluate the performance of a matching algorithm. This is usually done by using the standard information retrieval metrics of *precision, recall* and *f-measure* [4].

- *Precision:* measures the ratio of correct correspondences over the total number of returned ones. It reflects the degree of *correctness* of an algorithm.

$$precision = \frac{\#true\_positives}{\#correspondences\_found} \quad (1)$$

- *Recall:* measures the ratio of correct correspondences over the number of expected ones. It reflects, the degree of *completeness* of an algorithm.

$$recall = \frac{\#true\_\text{positives}}{\#existing\_\text{correspondences}} \quad (2)$$

- *F-measure:* is a measure introduced to compare the systems with just one value since it is highly likely that the system with a higher recall may have a lower precision and vice versa.

$$f - measure = \frac{precision * recall}{(1-\alpha) * precision + \alpha * recall} \quad (3)$$

These measures were used to evaluate the performance of our algorithm. Among the range of tests at the OAEI-13 we have tested our algorithm with several of them, although for the scope of this paper we will be focussing on the *conference* track which aims at finding alignments within a collection of ontologies from the domain of conference organization. The results obtained by our algorithm for each pair of input ontologies are compared with a reference alignment also obtained from the OAEI-13 website. In table 1 we include the average results obtained for this task.

Table 3. Average values obtained in the conference track

Precision	Recall	F-measure
0.86	0.57	0.67

In the smart cities domain, there is a series of ontologies that describe the resources and services that are available for the agents. If an agent needs a certain resource or service, it will need to match its ontology to the appropriate one in the smart city. Depending on where the service or resource is deployed, the agent may need to match its ontology to a part of the ontology that describes the smart city itself, usually when the agent needs access to a resource, or to another agent's ontology, usually when the agent needs a service that is offered by the other one. This situation is depicted in figure 2. In any case, this process will output an alignment between both ontologies. If several agents need to access the same resource or service, the process will be repeated several times.

Our intuition is that if a new agent arrives in the smart city and is willing to use a service or resource, the alignments previously obtained from other agents may help in tuning the alignment process for this new agent and therefore they may be used to enhance the results produced by the algorithm.

This led us to delving into *alignment reuse techniques* [20] which in spite of not being a particularly used matching technique [4], it was precisely the one that better met the our requirements. This technique is grounded on the idea that when describing an application domain the ontologies to be matched are similar to already matched ones and hence this knowledge may be reused. This idea was implemented in the *COMA* [21] and *COMA++*[22] systems which are two of the most well-known ones and that have been continuously evolving since 2002 to include new matchers and features.

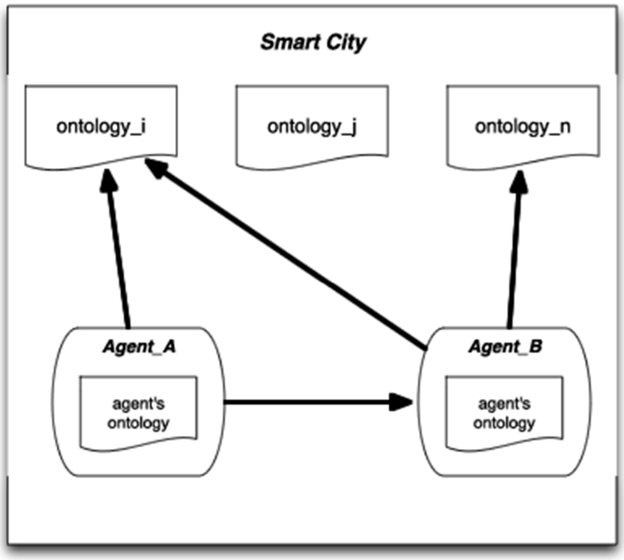


Fig. 2. **Fig. 3.** Smart City

To asses the viability of integrating alignment reuse as part of the ontology matching proposal for smart cities, we have used the ontologies of the conference track. The procedure followed to do so is to feed the algorithm with some intermediate alignments that are then used to identify binding points between the ontologies to match.

Consider the following example, let us suppose that there are three different ontologies, A, B and C, and that we need to match ontology A to ontology C. If we also have available the alignments between A and B ($Align\_A\_B$) and, B and C ($Align\_B\_C$), then it is possible to identify a path that, using these intermediate alignments, may link entities in A to entities in C. We refer to this as a *ring* between A and C through B, and it is graphically represented in figure 3.

Tables from 2 to 6 show the results of testing this approach with the ontologies of the conference track. From the ontologies available at this track in the following examples we have used the following ones: *cmt*, *conference*, *confOf*, *edas*, *ekaw* and *sigkdd*.

Table 2 shows the results obtained by directly matching the *conference ontology* to the *confOf ontology*. These values are included to provide a baseline to compare the results obtained when using rings.

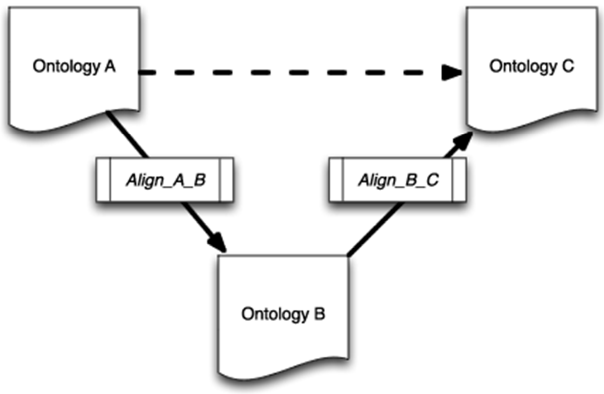


Fig. 3. **Fig. 4.** Ring

Table 4. Results obtained without using any ring

conference - confOf	
Precision	0.9
Recall	0.6
F-measure	0.72

Tables 3 and 4 show two different sets of results obtained when using an additional ontology as ring. Table 3 contains the set of results obtained when matching *conference* to *confOf*, using as additional input the alignments output when matching *conference* to *edas* and *edas* to *confOf*. Table 4 presents the set of results obtained using *ekaw*. As we can observe, in any case, the values obtained are better than those in table 2. However, the improvement using *edas* was more noticeable in the *recall*, while the improvement using *ekaw* was in the *precision*.

Table 5. Results obtained using *edas* for the ring

conference - (edas) - confOf	
Precision	0.86
Recall	0.80
F-measure	0.83

Table 6. Results obtained using *ekaw* for the ring

conference - (ekaw) - confOf	
Precision	1.00
Recall	0.73
F-measure	0.84

When using *edas* for the ring, 10 different paths from *conference* to *confOf* were detected which allowed the identification of 5 new correspondences. Using *ekaw*, just 8 different paths were identified which added 2 new correspondences that were not detected when directly running the matching process. We considered then a combined approach using both *edas* and *ekaw* at a time, seeking to obtain results with the precision enhancement provided by *ekaw* and the recall enhancement provided by *edas*. The results obtained are shown in table 5.

Using a multiple ring the amount of identified paths rises to 13. The results obtained with this approach show that precision is not as high as when just using the single ring with *ekaw* as there is an extra incorrect correspondence that is added to the final output. However, for recall and F-measure, the values obtained are remarkably higher that those obtained with single rings and when compared to the baseline results we observe an improvement of 10.95% for precision, 62.26% for recall and 35.48% for f-measure.

Table 7. Results obtained with *edas* and *ekaw* used at the same time

conference - (edas &ekaw) - confOf	
Precision	0.92
Recall	0.86
F-measure	0.89

In spite of being a positive outcome, the results vary when using the alignments with other ontologies as rings. Table 6 shows the results obtained when considering the alignments with *cmt* and *sigkdd* for the rings. The results in this table also show an improvement compared to the baseline in table 2, although they are not as remarkable as those in table 5.

Table 8. Results obtained with *cmt* and *sigkdd* at the same time

conference - (cmt & sigkdd) - confOf	
Precision	0.90
Recall	0.66
F-measure	0.76

Other tests run using more alignments showed no improvement compared with using just two as in the examples presented previously. However the testbed that we used is not large enough to entirely dismiss that possibility. An issue that we have identified is that even when using the reference alignments provided by the OAEI, which are the golden standard used to compare the results of any algorithm, there were some paths that we identified, that led to selecting as binding points pairs of entities that then were not considered as a valid correspondence in the reference alignment.

4 Conclusions & Future Work

In this paper we have introduced the smart cities domain and underline some communication and interaction problems that are highly likely to show up in this kind of development. We have also described the foundations of the *ontology matching* based approach that we propose to tackle such problems in the smart cities, and the ontology matching algorithm that we have defined to address this problem. We have described the measures of precision, recall and f-measure used to evaluate this type of algorithm, and the results obtained when doing so.

We have then looked at some alternatives to refine the alignments obtained by our algorithm and therefore improve such results. Among the different techniques we have focused on *alignment reuse* as it was particularly indicated for scenarios like ours. We have used previously existing alignments, *rings*, to identify paths of between the ontologies to match and therefore enhance the results obtained. We have tested this approach using the ontologies available from the *conference track* of the Ontology Alignment Evaluation Initiative 2013. And we have verified the viability and validity of the approach.

In spite of these good results that account for the viability of our approach, there are some issues that need to be addressed in order to obtain the best results possible and hence to improve the usability of the smart cities. There is, for instance, the need to test our proposal using ontologies taken from the real domain where it will be deployed, the smart cities. Additionally, as we introduced in section 3, there seems to be a direct relation between the rings chosen and the goodness of the results obtained, so we aim at focusing on determining the features that make a *ring* better than other. It is also necessary to explore techniques that will allow the alignment reuse in the real environment, so we are turning to some techniques such as *alignment storing and sharing* and *alignment annotation*.

References

[1]Katole, B., Sivapala, M., V, S.: Principle elements and framework of internet of things. International Journal Of Engineering And Science **3**(5) (07 2013)

[2]Schaffers, H., Komninos, N., Pallot, M., Trousse, B., Nilsson, M., Oliveira, A.: Smart cities and the future internet: Towards cooperation frameworks for open innovation. In Domingue, J., Galis, A., Gavras, A., Zahariadis, T., Lambert, D., Cleary, F., Daras, P., Krco, S., Müller, H., Li, M.S., Schaffers, H., Lotz, V., Alvarez, F., Stiller, B., Karnouskos, S., Avessta, S., Nilsson, M., eds.: Future Internet Assembly. Volume 6656 of Lecture Notes in Computer Science., Springer (2011) 431–446

[3]Chin, J.S.Y., Callaghan, V., Clarke, G., Hagras, H., Colley, M.: End-User Programming in Pervasive Computing Environments. In Yang, L.T., Ma, J., Takizawa, M., Shih, T.K., eds.: PSC, CSREA Press (2005) 187–192

[4]Euzenat, J., Shvaiko, P.: Ontology matching. 2nd edn. Springer-Verlag, Heidelberg (DE) (2013)

[5]Uceda-Sosa, R., Srivastava, B., Schloss, B.: Using Ontologies to make Smart Cities Smarter. IBM Research. (06 2012)

[6]:SOFIA2. http://scfront.cloudapp.net/ (5 2014)

[7]Wiesman, F., Roos, N., Vogt, P.: Automatic ontology mapping for agent communication. In: Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2, ACM (2002) 563–564

[8]Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. IEEE Transactions on Knowledge and Software Engineering (2012)

[9]Gal, A., Shvaiko, P.: Advances in ontology matching. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **4891 LNCS** (2008) 176–198

[10]Cohen, W.W., Ravikumar, P.D., Fienberg, S.E.: A Comparison of String Distance Metrics for Name-Matching Tasks. In: IIWeb. (2003) 73–78

[11]Duman, H., Hagras, H., Callaghan, V.: Intelligent association exploration and exploitation of fuzzy agents in ambient intelligent environments. Journal of Uncertain Systems **2**(2) (2008) 133–143

[12]Rahm, E., Bernstein, P.A.: A Survey of Approaches to Automatic Schema Matching. Very Large Data Base (VLDB) Journal **10**(4) (2001)

[13]Jiménez-Ruiz, E., Grau, B.C. In: Logmap: Logic-based and scalable ontology matching. Springer (2011) 273–288

[14]Hanif, M.S., Aono, M.: Anchor-flood: Results for oaei 2009. In Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H., Noy, N.F., Rosenthal, A., eds.: OM. Volume 551 of CEUR Workshop Proceedings., CEUR-WS.org (2008)

[15]Noy, N.F., Musen, M.A.: Anchor-prompt: Using non-local context for semantic matching. In: Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001), Seattle, WA (2001)

[16]Le, B.T., Dieng-Kuntz, R., Gandon, F.: On Ontology Matching Problems (for building a corporate Semantic Web in a multi-communities organization). In: Proc. of the Sixth International Conference on Enterprise Information Systems - Porto - Portugal, Kluwer (14-17 April 2004)

[17]Miller, G.A.: WordNet: a lexical database for English. Communications of the ACM **38**(11) (1995) 39–41

[18]World Wide Web Consortium. W3C: OWL: Web Ontology Language (2014)

[19]Euzenat, J., Meilicke, C., Stuckenschmidt, H., Shvaiko, P., dos Santos, C.T.: Ontology Alignment Evaluation Initiative: Six Years of Experience. Journal on Data Semantics - Springer (2011) 158 – 192

[20]Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. The VLDB Journal **10**(4) (2001) 334–350

[21]Do, H., Rahm, E.: COMA - a system for flexible combination of schema matching approaches. In: Proceedings of the 28th VLDB Conference, Hong Kong, China (2002)

[22]Aumueller, D., Do, H.H., Massmann, S., Rahm, E.: Schema and ontology matching with COMA++. In: Proc. ACM SIGMOD Conference. (2005)

[23]Naphade, M.R., Banavar, G., Harrison, C., Paraszczak, J., Morris, R.: Smarter Cities and Their Innovation Challenges. IEEE Computer **44**(6) (2011) 32–39

[24]Uceda-Sosa, R., Srivastava, B., Schloss, R.J.: Building a Highly Consumable Semantic Model for Smarter Cities. In: Proceedings of the AI for an Intelligent Planet. AIIP ’11, New York, NY, USA, ACM (2011) 3:1–3:8

[25]Villanueva, F.J., Santofimia, M.J., Villa, D., Barba, J., López, J.C.: Civitas: The Smart City Middleware, from Sensors to Big Data. In Barolli, L., You, I., Xhafa, F., Leu, F.Y., Chen, H.C., eds.: IMIS, IEEE (2013) 445–450

[26]Fazio, M., Paone, M., Puliafito, A., Villari, M.: Heterogeneous Sensors Become Homogeneous Things in Smart Cities. In You, I., Barolli, L., Gentile, A., Jeong, H.D.J., Ogiela, M.R., Xhafa, F., eds.: IMIS, IEEE (2012) 775–780

Interpreting Service Using an Upper-level Ontology

Anders W. Tell,

Stockholm University
Department of Computer and Systems Sciences, Sweden
{anderswt}@dsv.su.se

Abstract. The emergence of service analysis, design, architectures and solutions presented in service marketing and software engineering literature has created a need for understanding the nature of services. Services are often considered as possessing characteristics that are assumed to pose specific problems for service providers as opposed to providers of goods. This paper presents an ontological interpretation of the concept of service using a general and upper level ontology with a strong base in natural sciences. The Basic Formal Ontology (BFO) is used to interpret the concept of service, as defined in the Service Dominant Logic approach. The interpretation is demonstrated in an analysis of service characteristics, in relation to goods. The ontological and reductionist approach opens up to a formulation and analysis of service, a social and economical phenomenon, in terms of general natural science oriented concepts. The ontological grounding provides a language that supports alignment of specific service definitions used in different subject fields, as well as alignment with adjacent concepts such as capability. The interpretation and analysis support the conclusions that studied characteristics are relevant to the concept of service, although they cannot be considered as determinant characteristics of service, and from a practical point of view they contribute partially to observed concerns and problems.

Keywords: Service, Ontology, Basic Formal Ontology, Service Dominant Logic.

1 Introduction

This paper revisits the concept of *service* and explores a novel kind of ontological interpretation that enables a reductionist analysis of the nature of services, its similarities with and differences from goods.

The distinction between, and duality of goods and services have been a vibrant topic of economical, market, and software engineering research and practices. The word 'service' has been part of natural language since 13th century and subsequently carries substantial common sense meaning. The topics of services and goods have been discussed amongst economists [1], such as by Adam Smith in his book, The Wealth of Nations, from 1776, and the discussions continues even today.

The search for defining and distinguishing characteristics of service have constituted a key research topic, however practitioners [2] and researchers [3],[4] find that characteristics such as intangibility, inseparability, heterogeneity, perishability does not constitute major service problems and are not determining criterion of services. The on-going debate illustrate that a deeper understanding of the concept of service is needed.

This paper address the question if it is possible to create a formulation, interpretation, of the concept of service, using a higher level ontology, that can be used to explain the differences in argumentation relating to service characteristics, relations between services and goods, and practical consequences of servicing.

This question is part of a design science inquiry into the requirements that influence the design of a knowledge organisation construct, Ability Perspective, an (Enterprise) Architecture Viewpoint [5]. In the design of an Ability Perspective it is important that knowledge about capabilities [6], and abilities are aligned with related, adjacent concepts such as service since service is sometimes considered as a mechanism to enable access to a set of one or more capabilities [7]. Furthermore, the use of an upper-level ontology provides an ontological grounding, a language, that support alignment of specific service definitions used in different subject fields, e.g. organisational design, marketing and information technology (IT).

The aim of this paper is to present of an ontological interpretation of the concept of service in terms of general concepts. The upper-level ontology - Basic Formal Ontology (BFO) [8] - is used to interpret 'service' as defined by Vargo and Lusch in their Service Dominant Logic (S-D Logic) work [9, 10]. The interpretation is explored and demonstrated in an analysis of service characteristics and relations to goods. For brevity, this paper covers an analysis of the frequently cited characteristics; intangibility, inseparability, heterogeneity, and perishability.

The main contributions of this paper are firstly, a novel ontological interpretation of the concept of service using a upper-level ontology that provides a bridge between natural and social sciences, and offers clarifications of the constituent parts of the concept of service. Secondly, conclusions that analysed service characteristics are relevant but not determinant, and that practical implications of the service characteristics depend on specific kinds of services. Thirdly, the introduction, in section 3.2, of the *'lead-to pattern'* that provides a novel and flexible approach for informed reasoning about value creation along *Result Ladders* up to the *Service Horizon*.

The paper is structure as follows: In section 2 an outline of the objects of analysis, services characteristics and aspects, is presented. In the 3rd section, the theoretical basis is introduced. I section 4 an interpretation of the S-D logic service definition using BFO is presented. The services characteristics are then analysed in section 5. The paper is concluded with sections with future research and conclusions.

2 Services characteristics

In this section the service characteristics that are chosen as subject of interpretation and analysis are presented. They are well known, and often cited as vital and relevant in service marketing literature by scholars [2], [1], [4], [3]. These characteristics are argued to separate services from goods and that they constitute major cause for service concerns and problems that are different from goods concerns and problems.

The **intangibility** characteristic of service suggests that the performance, application of competences don't have material qualities and cannot be experienced, seen, heard, felt, smelled, or tasted. According to Zeithaml [2] service specific problems include: services cannot be stored, services cannot be protect through patents, cannot readily display or communicate services, and prices are difficult to set.

The **inseparable** characteristic suggests that both a producer and consumer that must be both present at the time of performance of the service; i.e. a service is produced, delivered and consumed simultaneously, and centralised mass production of services is difficult [2].

The **heterogeneity** (inconsistency, variability) characteristic relates to the variability, inconsistency, of a service performance. A service may be rendered differently over time and space, and some qualities may vary across service producers, e.g. a person could by tired one day and well rested another day, standardisation and quality control are difficult to achieve [2].

The **perishability** (inventory) characteristic relates to that a service, a performance of services, or application of competences, cannot be stored for future reference, delivery, or use [2].

3 Theoretical basis

The theoretical basis consists of three parts; the subject of interpretation: a service definition from Service Dominant Logic, an upper level ontology - Basic Formal Ontology, and additional analytical tools.

For clarity the names, designations of previously defined concepts are prefixed with an abbreviation indicating the ontological domain they belong to: sdl: Service Dominant Logic, bfo: Basic Formal Ontology, ext: Analytical tools, extensions.

1.1 Services according to Service Dominant (S-D Logic)

As the subject of interpretation and analysis the service definition from Service Dominant Logic by Stephen L. Vargo and Robert F. Lusch [9] is selected because it is well known, contemporary, and consists of relatively few parts, thus making interpretation, reduction, reformulation and analysis feasible.

The Service Dominant Logic is an approach, perspective, mind-set, and theory about the nature of *service*, in relation to goods, within the realm of marketing and economic exchange. S-D Logic offers an alternative to the prevailing dominating goods perspective. Instead of focusing on the exchange of goods the focus should shift to a focus to the value that various activities, material and immaterial entities provide, i.e. to the service they provide.

In S-D Logic, service is defined as the process of using one’s competences (knowledge and skills) through deeds, processes and performances for the benefit of another part.

Alternatively a service focus may be formulated in terms of *operants* that operate on *operands* (goods, material and immaterial entities, and other services).

3.1 Basic Formal Ontology (BFO)

The Basic Formal Ontology is an upper-level ontology that supports the creation of lower-level domain specific ontologies.

The BFO project [8] started in 2002 with initial theoretical contributions from Barry Smith and Pierre Grenon. The aim of BFO is to provide a genuine upper ontology that specifically can be extended by domain ontologies developed for scientific research, such as for biomedicine. BFO is based on the principle of ontological realism [11], where ontologies are viewed as representations of the reality that are described by science.

Key elements of BFO are the support for formal (logical) reasoning enabled by its definition and the inclusion of common formal theories such as mereotopology and qualitative spatial reasoning. As part of the effort to formalize BFO, the BFO is defined using OWL and in first order logic using the CLIF (Common Logic Interchange Format) from ISO.

With respect to other public domain ontologies DOLCE, SUMO and CYC, BFO aims at, and provides a smaller core that is extendable and adaptable to specific domains [11], thus making it suitable for creation of a service specific extension. BFO shares some philosophical basis with DOLCE and SUMO, such as the inclusion of 'universals' and 'particulars' as well as the acceptance of a dichotomy between 'continuants' ('endurant') and 'occurrents' ('perdurant').

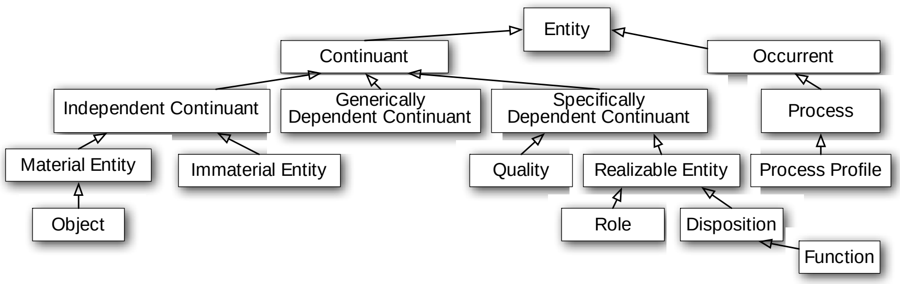


Fig. 4. Illustration of key concepts from the BFO Ontology, version 2012 July

Table 9. Description of BFO concepts used in the interpretation

BFO	**Description**
Entity	An entity is anything that exists, or has existed, or will exist
Continuant	An entity that persists, endures, or continues to exist through time while maintaining its identity. ('endurant')
Occurrent	An entity that unfolds itself in time, or it is the instantaneous boundary of such an entity (for example a beginning or an ending), or it is a temporal or spatiotemporal region, which such an entity occupies. ('perdurant').
Material Entity	Has some portion of matter as proper or improper continuant part. ‘Portion of matter’ is intended to encompass both mass and energy. Every material entity at any given time is localized in space at that time, and can move in space. Material entities are three-dimensional spatial entities, as contrasted with the processes in which they participate, which are four-dimensional entities. Example: a human, an aggregate of humans.
Object	A material entity, which manifests *causal unity* via physical covering (organisms,

	cells), or internal physical forces (portions of solid matter such as rocks and lumps of iron), or engineered assembly of components (engineered artifacts such as watches and cars). Objects can be joined to other objects and may include other objects as parts. Examples: cell, organism, grain of sand.
Immaterial Entity	Have no material entities as parts. Examples: surface, line, point
Process	Has temporal proper parts and for some time t, P s-depends-on some material entity at t. *has-participant* is an instance-level relation between a process, a continuant, and a temporal region at which the continuant participates in some way in the process. A process do not change, it is the change itself. Examples: the life of an organism, a process of sleeping.
Process Profile	A process that represents a selective cognition or abstraction of mutually dependent sub processes. Examples: a pair of rumba dancers is moving together across the dance floor form a mutually dependent process pair, the process of temperature changes in John

3.2 Analytical tools

For the purpose of analytical convenience we introduce 6 supporting concepts: Performer, Servicing, Result, Lead-to pattern, Result Ladder and Result Horizon.

A **Performer** is a *bfo:Material Entity* that can change the world. The *ext:Performer* concept facilitates an understanding of questions relating to interrogative 'who', 'who is doing what' and *sdl:Operants*. Examples include; Natural performers (organisms, humans,...), Man Made (machine, information system,...), Social (person, organisation unit, enterprise, ...), and Roles (actor, worker, ...).

The word "service" is often used as a sign for both the act of 'applying competences' and the resulting value, benefit part of (a) service(s). **Servicing** is a *bfo:Process* that specifically corresponds to the act of applying competences.

The following *Lead to pattern*, *Result Ladder,* and *Result Horizon* concepts provides the primary vehicles for reasoning about transfers of benefits from providers to beneficiaries.

The general three-part **Lead to pattern** is a pattern where *some source entities lead to some result entities* [6]. This pattern is pervasive in science, theories and frameworks, e.g. causality - effect from cause, means to some ends, marketing - attributes lead to consequences that lead to values [12] and templating [13].

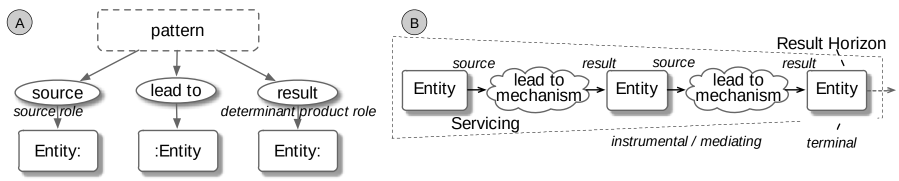


Fig. 5. Illustration of (A) Lead-to pattern, and (B) Result Ladder and Result Horizon concepts

- A ***Source*** *entity* participates in a thematic source role, e.g. source entities are instrumental in bringing about a result. Examples: material and immaterial entities, humans, competence, skills, knowledge, information, and performers.

- A ***Result*** *entity* participates in a thematic determinant product role. Examples include; value; benefit; satisfaction of objective; and entity, quality, functioning that comes into being, dies, does not come into being; and a state-of-affair; and a change or no change; and some act that is completed-done-not done.
- The ***Lead-to*** *entity* is an entity that provides a link between the source and the result, where results are lead to, brought about, achieved, accomplished, realised, made, generated, etc. Examples of lead-to mechanisms include; a realising process, mechanism, causality, logical entailment, counter factual specification, probabilistic specification, and mathematical formulas.

A **Result Ladder** is a partially ordered set of *ext:Results* with *ext:Lead-to* links in-between, where a *ext:Result* may play the role of a *ext:Source entity* for following *ext:Lead to* links. A ext:Result Ladder may include intermediary *ext:Lead to* links that *mediate* a transfer between *source entities* and terminal *result entities* over time and space.

In marketing and in services research numerous examples of ladders are found, e.g. Means-End Theory where product attributes (A) lead to consequences in product use (C), to individuals' values (V) [14], value theories [15] such a Rokeach instrumental and terminal values, and Cocktons worth maps [16]. In the analysis no specific value theory is assumed.

The *ext:Result Ladder* concept enable a detailed analysis of a number of benefit related aspects and questions: What is valued, which value is attributed to whom? Where and when are *ext:Results* observed and measured? How many *ext:Result Ladders* exist simultaneous (customer, provider, worker, owner, society, ...)? Is there a single terminal end point, or multiple? If so, does the value ladder terminate in some universal value space ("everything"), or at some value attributed to some single entity ("the"), or in societal values ("we"), or in experiential values of (all) sentient beings ("i"), or in some values that evolve over time?

A **Result Horizon** specifies the time, space and end result(s) of *ext:Result Ladders*. As such it is analogous to an investment horizon.

The ext:*Result Horizon* concept enables a detailed analysis of questions and aspects that influence design, management and governance of services: Where does a *ext:Result Ladder* ends, or should end? Is it at servicing completion, or at the exchange of service performance for money, or should/must the horizon be longer? Higher order values such as dignity, justice or gender equality may be assumed to be even (qualitative) better than money, or emotional values.

4 BFO based interpretation of the S-D Logic service definition

In this section the interpretation of the service definition using the Basic Formal Ontology (BFO), is presented. The following S-D Logic definition constitute the base for interpretation and subsequent analysis:

> "the application of competences for the benefit of another" [10].

The service definition involves two agentive entities that play two roles, the **applyer** (producer, provider) and, **another** (consumer, beneficiary). From a systems perspective the two entitles may be considered as two (hard or soft) systems [17].

In the following diagram the rectangles represent concepts, the ovals represent concept relations, and the lines represent arguments of a concept relation.

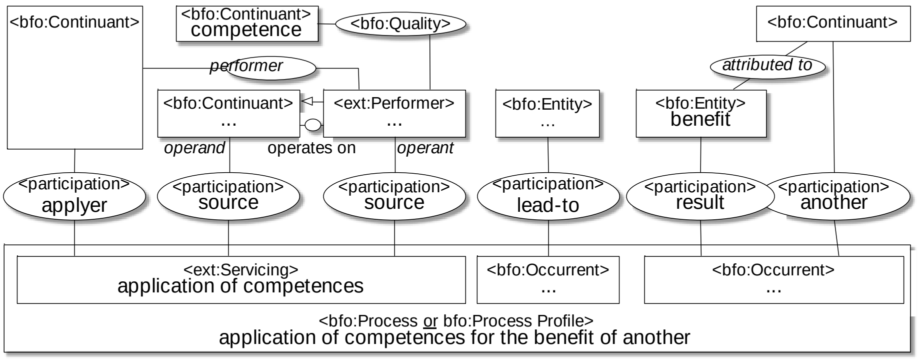


Fig. 6. Illustration of the BFO based interpretation of S-D Logic service definition

The "**application of competence**" is interpreted as consisting of five constructs:

1. The **applyer** entity that is attributed to the 'application of competence for the benefit of another', is interpreted generally as a *bfo:Continuant*. For specific kinds of service a more suitable subtype may be defined, e.g. a Human being can be represented as a *bfo:Material Entity*.
2. An *applyer* incorporates of at least one entity that can **perform** the "application of competences", i.e. change the world and deliver benefits. This entity is interpreted as a *ext:Performer,* that is competent (have requisite or adequate ability or qualities). In many cases the *applyer* and *performer* are the same entities, e.g. a human. In other cases they are different, e.g. an organisation consist of persons performing the servicing.
3. **Competences** are for the purpose of brevity interpreted as *bfo:Continuant* qualities of a *ext:Performer.*
4. The S-D Logic service definition do not explicitly include references to entities other than competences it is however implicitly understood in S-D Logic that more entities can participate in a *ext:Servicing* process, such as material and goods. These **additional entities** are added to the interpretation as participating *bfo:Continuants*.
5. The **application of competences** occurrence is interpreted as a *ext:Servicing process* in which the *applyer* participates, together with at least one *ext:Performer* and possibly one or more *bfo:Continuant* in source roles.

The "**the benefit of another**" is interpreted as consisting of three constructs:

1. An **another** entity that is interpreted as a *bfo:Continuant.*
2. The **benefit** entity is interpreted as an *bfo:Entity* attributed to the *another* entity.

3. Both *another* and the *benefit* entities participate in an *bfo:Occurent* where the benefit comes (or not) into being. The *benefit* entity participates in a *result role* and the *another* entity in an *another role*.

The last part of the definition to interpret is the "**for**" part that links the "application of competences" with the "benefits of another". The separation of the two (or more) agentive entities implies that some form of transfer of results, must exist, directly or through intermediary entities and/or over time and/or space. Here the *ext:Lead to pattern* is applied to represent linking and transfer aspects between *applyer* and *another*. The competent *ext:Performer*, and additional entities attributed to the *applyer* corresponds to entities playing the *source role*, the *benefits* attributed to *another* corresponds to *result role entities*, and the overall *application of competences for the benefit of another* process corresponds to the *ext:Lead to* part.

Depending the specific kind service defined or analysed, the overall "application of competence…" *bfo:Process* can be represented as one large *bfo:Process*, or as a group of smaller *bfo:Processes*, or as a *bfo:Process Profile* in alignment with soft systems thinking [17].

For a specific kind of service it is possible to consider both the *applyer* and *another* as causally united *bfo:Objects* and then associate a *bfo:Process* with each, possibly together with *intermediary bfo:Occurrents*, or *bfo:Processess*, that mediate the transfer of benefits. This kind of separation opens up to *ext:Result Ladder* and *ext:Result Horizon* reasoning.

A **Service horizon** is defined as a *ext:Result Horizon* that determines the scope of service benefits *ext:Result Ladder(s)* to consider. It should be noted that the *applyer* and *another* entities may value intermediary results and terminal benefits differently.

The **Triple-O** service constructs: *sdl:Operant* is interpreted as a *ext:Performer*, and *sdl:Operant* as additional *bfo:Continuants* participating in a *bfo:Occurent*.

5 Analysis of a interpreted service definition

This section presents an analysis of service characteristics based on the Basic Formal Ontology based interpretation of service as defined in S-D Logic. The analysis focuses on exploring the following questions: Is the characteristic **relevant** to the definition of the concept of service? Is the characteristic **determining**/distinguishing service from goods? Does the characteristic impact **practical** considerations?

5.1 Intangibility

In the service interpretation three sources of *intangibility* are identified:

- *bfo:Occurrent*, the occurrence of "application of competences.", e.g. hair cutting.
- participating source intangible *bfo:Immaterial Entities*, e.g. hair style.
- resulting benefit *bfo:Entity* at the end of a *Service Horizon*, e.g. customer value.

In the service interpretation *Tangible* phenomenons are identified:

- *ext:Performers*, *sdl:operants*, e.g. hairdresser and customer.
- *bfo:Material entities*, *sdl:operands*, e.g. chemicals, pair of scissors and hair.

What could be argued is that goods, in an opposite sense, can be seen and tasted, e.g. a cake, hair cut. However some goods are intangible; e.g. hairstyle, news, patents and intellectual properties.

The relevance of tangibility depends on one or more *Service horizons*, e.g. an intangible hairstyle, lead to a tangible hair cut, which leads to emotional intangible values at a later dinner. One argument for intangibility is that all *Service horizons* finally end up in sentient beings or humans with resulting intangible terminal emotional values. This argument is problematic, from a practical point of view, since service agreements often are specified in terms of shorter *Service horizons* and tangible results, such a cut hair or other functional qualities. Furthermore the actual *ext:Result Ladders* and *Service horizons* relevant to the provider, and beneficiary are in general different.

From a practical point of view, the qualities of a service depend on the mix of participating tangible *bfo:Performers,* and *bfo:Continuants (material or immaterial).*

Goods produced in manufacturing processes can be viewed as the sum of processing steps, or a historical embodiment of *ext:Servicing* [18]. The last production steps can be viewed as embedded services performed by the supplier on behalf of the consumer.

Based of the analysis I find that intangibility is a relevant characteristic of a service, although not a determinant characteristic. From a practical point of view intangibility is relevant, however a focus on intangibility may obfuscate the relevance of tangible entities along *ext:Result Ladder* and at the *Service horizon.*

5.2 Inseparability of production and consumption

In a hairdressing service case, a hairdresser (*applyer*) and consumer (*another*) meets, rendezvous directly in several *bfo:Processes*, 'wash hair', 'modify hair', etc. where the application of competences lead to benefits (cut hair) for the consumer, i.e. a direct transfer *ext:Lead to* mechanisms are present. However benefits may materialise at a later stage, in some other location, possibly at a dinner, mediated by the hair and the mental state of the consumer. The *ext:Result Ladder* (and *Service horizon*) may be longer than the time and space where hairdresser and consumer rendezvous, or the period of *ext:Servicing,* or the duration of a service agreement.

When buying a tree in order to facilitate clean air for our children; the transfer (lead-to) processes between the tree buyer, planter and future beneficiaries are many and separated by time and space.

Based on the analysis I find that inseparability is a relevant characteristic of service, although not a determinant characteristic. From a practical point of view, reasoning about inseparability could be replaced by reasoning about *ext:Lead to* processes, *ext:Result Ladders* and *Service horizons.*

5.3 Heterogeneity (inconsistency, variability)

Two separate services, applications of competence, implies two distinct *bfo:Processes*, leading to differences in the quality. Three sources of variability in servicing are identified in participating *bfo:Continuants*:

- *ext:Performer*, e.g.hairdresser, numerically controlled machine, computer.
- *bfo:Material entity*, e.g. pair of scissors, shampoo.
- *bfo:Immaterial entities*, e.g. hair style.

In goods production processes the qualities of participating *ext:Performers* have an impact on the variability in the qualities of goods. Labour intensive production processes involve variability patterns that differ from processes executed by high quality man-made machines, or computers. In cases where the supply of goods are demand driven and engineered-to-order, the production process may involve higher variability due to re-tooling, configuration of machines, and human creative activities that cannot fully benefit from being repeatable and standardised.

Analogous to production of goods, the same *ext:Performer* and *bfo:Material entities* may participate in two different *ext:Servicing processes*, which may lead to lower expected variability. Variability in service availability may be mitigated by sourcing and dynamic capacity management of participating *bfo:Continuants*.

In the case of a custom-made toy making service, a company may choose to manufacture toys with high performance man-made machines, using steel as material, or human craftsmen that carves toys out of wood. Here an interesting questions arise, what is more relevant, the variability of servicing or the variability of the participating material? Depending on how variability is measured a highly variable, man made, old and ragged, toy made out of wood may be considered as more (emotionally) valuable than a durable, hard, cold and low variable steel toy.

Based on the analysis I find that heterogeneity is a relevant characteristic of service, although not a determinant characteristic. From a practical point of view all sources of variability should be considered along relevant *ext:Result Ladders*.

5.4 Perishability (inventory)

In the service interpretation four sources of perishability are identified:

- *bfo:Process* ("application of competences"), e.g. cutting of a hair,
- *ext:Performer*, e.g. hair dresser that gets older and forgets,
- source *bfo:Entities*, e.g. shampoo and pair of scissors that degrades over time.
- resulting *bfo:Entity*, e.g. emotional values of a haircut that diminish over time.

An “application of competences” (*bfo:Process*) cannot be stored for future use. On the other hand, *ext:Performers* and participating *bfo:Material entities* (pair of scissors, rental facilities) can be stored, and they can be acquired or produced in advanced for later participation in delivery. Neither *ext:Performers* and *bfo:Material entities* need to be owned or controlled. In a sourcing scenario they can be accessible (in mint condition) and thus replace inventory.

Goods (interpreted as *bfo:Material entities)*, as well as *ext:Performers* and *bfo:Material entities* that participate in manufacturing processes may perish over time; people and machines get old, material decompose, chemicals degrade, etc. *bfo:Immaterial entities* such as hair styles, songs, digital and reproducible material exist over time without diminished capacity, although they may be forgotten.

Based on the analysis I find that perishability is a relevant characteristic of service, although not a determinant characteristic of service. From a practical point of view the temporal and inventory aspects of participating *ext:Performers*, *bfo:Material entities* and *bfo:Immaterial entities* are highly relevant for both services and goods.

6 Future Research

Through a formalization of the service definition, using the same first order logic construct used in the formalisation of BFO, a domain specific service extension to BFO can be created. Such formalisation provides a platform for detailed analytical comparisons between a wider range of service definitions, service aspects, such as *co-creation of value*, and inquiries into larger systems based on services and goods.

The use of BFO as an ontological grounding (language) can be used to integrate a service construct with adjacent constructs such as capability. In the paper "What Capability Is Not" [6] I provide a conceptualisation of a Capability construct, "a substantial possibility that source entity(ies) lead to a result", based on the *ext:Lead to* pattern, that integrates well with presented service interpretation.

The BFO interpretation together with the Triple-O concepts (operant, operand, occurrence) suggests that a simplified, socially oriented, definition of service may be possible, e.g. "use of effort (energy, material) leading to a result of another".

7 Conclusions

In this paper, I have revisits the concept of service and presented an interpretation of the concept of service as defined by the Service Dominant Logic framework, using a higher level ontology - Basic Formal Ontology. The use of BFO enables an inter-theoretical reduction and a bridge between social and natural sciences.

Based on the interpretation and analysis I argue that the use of BFO provides clarifications, of the constituent parts of the concept of service (as defined in S-D Logic), and of similarities and differences between goods and services. However the interpretation and analysis of a single service definition reduces the possibility for making more general conclusions.

Furthermore I conclude that the studied characteristics are relevant to the concept of service, although they cannot be considered as determinant characteristics. From a practical point of view the studied characteristics contribute *partially* to observed concerns and problems.

In order to obtain a complete understanding for, both services and goods, it is necessary to consider the nature of all participating *bfo:Material Entities*, *bfo:Immaterial Entities*, and *ext:Performers* along one or more *ext:Result Ladders* up to the relevant or desired *Service horizon*.

These findings are consistent with studies [2], and the view that a service involves a *service perspective* on value creation rather than a category of market offerings [19].

References

1. Hill, P.: Tangibles, intangibles and services: a new taxonomy for the classification of output. The Canadian Journal of Economics/Revue canadienne d'Economique. 32, 426–446 (1999).
2. Zeithaml, V.A., Parasuraman, A., Berry, L.L.: Problems and strategies in services marketing. The Journal of Marketing. 33–46 (1985).
3. Araujo, L., Spring, M.: Services, products, and the institutional structure of production. Industrial Marketing Management. 35, 797–805 (2006).
4. Goldkuhl, G., Röstlinger, A.: Beyond goods and services: an elaborate product classification on pragmatic grounds. (2000).
5. ISO, IEC, IEEE: ISO/IEC 42010 Systems and software engineering — Architecture description. (2011).
6. Tell, A.W.: What Capability Is Not. Presented at the The 13 th International Conference on Perspectives in Business Informatics Research July 3 (2014).
7. Group, O.M., Inc: Unified Profile for DoDAF and MODAF (UPDM). (2012).
8. Smith, B., Grenon, P.: Basic Formal Ontology (BFO), http://purl.obolibrary.org/obo/bfo.
9. Vargo, S.L., Lusch, R.F.: Evolving to a New Dominant Logic for Marketing. 1–17 (2003).
10. Vargo, S.L., Akaka, M.A.: Service-dominant logic as a foundation for service science: clarifications. Service Science. 1, 32–41 (2009).
11. Smith, B., Ceusters, W.: Ontological realism: A methodology for coordinated evolution of scientific ontologies. Applied Ontology. 5, 139–188 (2010).
12. Hughes, J., Kroes, P., Zwart, S.: A Semantics for Means-End Relations. 1–15 (2005).
13. Sowa, J.F.: Relating Templates to Language and Logic, http://www.jfsowa.com/pubs/template.htm.
14. Veludo-de-Oliveira, T.M., Ikeda, A.A., Campomar, M.C.: Laddering in the practice of marketing research: barriers and solutions. Qualitative Market Research. 9, 297–306 (2006).
15. Johnston, C.S.: Taylor & Francis Online :: The Rokeach Value Survey: Underlying Structure and Multidimensional Scaling - The Journal of Psychology - Volume 129, Issue 5. The journal of psychology. (1995).
16. Cockton, G., Kirk, D., Sellen, A., Banks, R.: Evolving and augmenting worth mapping for family archives. Presented at the BCS-HCI '09: Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology September (2009).
17. Checkland, P.: Soft systems methodology: a thirty year retrospective. Systems Research and Behavioral Science. 17, S11–S58 (2000).
18. Vargo, S.L., Lusch, R.F.: Service-dominant logic: continuing the evolution. Journal of the Academy of Marketing Science. 36, 1–10 (2008).
19. Grönroos, C.: Service logic revisited: who creates value? And who co-creates? European Business Review. 20, 298–314 (2008).