A Decentralized and Ontology-Based Approach to Infrastructure Monitoring

Florian Fuchs, Michael Berger

Siemens AG, Corporate Technology
Intelligent Autonomous Systems, CT IC 6
Otto-Hahn-Ring 6, 81739 Munich, Germany
{florian.fuchs.ext | m.berger}@siemens.com

Abstract: We introduce infrastructure monitoring as an application domain that demands decentralized system designs. This is motivated by the large scale of these systems, the heterogeneity of components and the large number of stakeholders involved. Decentralization aggravates the issue of semantic interoperability, which we propose to tackle with a system information model based on formal ontologies. As a foundation for designing ontology-based infrastructure monitoring systems, we present a layered functional architecture and illustrate potential deployment scenarios. In particular, we detail the decentralized realization of the reasoning layer and describe an algorithm for distributed evaluation of conjunctive queries over distributed knowledge bases. A short description of the prototype implementation and first evaluation results are provided.

1 Introduction

Physical infrastructures is a generic term used for energy networks such as gas and power transmission networks as well as transportation systems such as rail networks and road networks. These infrastructures are essential for the functioning of a society and economy and have to cope with constantly increasing demands. These include *reducing cost* of infrastructure operation while retaining a high level of safety and reliability. A promising approach is optimizing maintenance procedures by moving from today's preventive maintenance, which is carried out at predefined time interval, to condition-based maintenance, which adapts maintenance intervals to the actual condition of a component. Essential here is up-to-date and high quality information on the current condition of infrastructure components. A different goal is *increasing utilization* and efficiency of use, which is driven by global trends such as urbanization and demand for environmental sustainability: The targets set by the European Rail Research Advisory Council (ERRAC) for the European rail infrastructure in 2020 include doubling the relative freight/passenger market share and tripling the absolute freight/passenger market volume [ERR02]. This requires optimization of infrastructure operations and again relies on the availability of up-to-date and high quality infrastructure condition information as the basis for informed decision making. A third major trend in infrastructure management is that *increasing numbers of stakeholders* are involved and need to be coordinated. This is caused by the ongoing effort to privatize (previously state-owned) infrastructures and introduce competition. Still, there remain interdependencies between infrastructure parts operated by different organizations, which require inter-organizational information exchange [SLW07]. Again, up-to-date and high quality information on infrastructure condition is necessary. In conclusion, availability of this information is essential for coping with today's and future demands on infrastructure monitoring as it will provide the basis for decision support systems (DSS) that help infrastructure managers in optimizing maintenance and operations.

Many sensor systems are already in place today, which provide up-to-date data about the condition of particular infrastructure parts. Examples from the rail domain are inductive sensors at switch machines, laser scanners for checking track geometry, and force sensors at the trackside for measuring the damage a passing train causes to the track. In the power network domain there are transformer load sensors, temperature sensors at cooling stations, and sensors for insulating gas density at circuit breakers. What is still almost completely missing, however, is an integrated analysis of the actually available data in order to derive more reliable and refined information [MRS05,KT04]. Moreover, external data sources such as weather services, which provide valuable additional information, are not taken into account. One of the major reasons for this is that data sources are highly heterogeneous and are operated by different organizations. This currently severely hinders information integration.

This paper addresses this problem by proposing an ontology-based system information model and decentralized system architecture. Its particular contributions are threefold: (1) the application of Description Logics-based knowledge representation *and* reasoning to infrastructure monitoring; (2) the identification of necessary functional layers for realizing this idea; (3) and the design of a framework for query answering over distributed Description Logics-based data. The paper is structured as follows: Section 2 reviews related work. Section 3 analyzes application requirements and argues for an ontology-based approach. Section 4 introduces a functional system architecture and proposes a decentralized deployment. The decentralized realization of the reasoning layer is detailed in Section 5, where an algorithm for answering ontology-based conjunctive queries over distributed knowledge bases is developed. Section 6 describes the prototype implementation and first evaluation results. Section 7 concludes the paper.

2 Related Work

Current infrastructure monitoring systems are usually stand-alone solutions that focus on monitoring only one or few aspects of an infrastructure component such as circuit breaker operation in the power network domain [KRL05] and e.g. hot wheel and sliding wheel detectors in the railway domain [RB01,ST98,Lag07]. Significant advantages in monitoring quality are expected from integrating these monitoring systems. Projects such as "Checkpoint" [MRS05] aim at integrating sensor systems (e.g. hot box detector, flat wheel detector, dynamic scale and loading gauge detector) for realizing an overall train inspection system. Still, these approaches are limited to the anticipated set of sensor systems and anticipated use cases as they lack machine-understandable modelling of the semantics of the relevant information.

Several projects currently address this problem by adopting formal ontologies [SLW07,TK07,FSW05,RFP06]. Their main focus is on information modelling, while the required system architectures and reasoning procedures are not further investigated. This paper, however, is particularly concerned with designing a system architecture and reasoning procedure for wide-area infrastructure monitoring based on ontologies.

Also related to this work is the OGC Sensor Web Enablement (SWE) initiative [BPR06], which aims at providing integrated access to distributed and heterogeneous sensor systems. A comprehensive body of standards is available, but there are major gaps in semantic interoperability support as formal ontological structures are still missing. [MS06] presents an ontological extension and application of SWE in the domain of wild fire detection. Due to the adoption of a multi-agent architecture, though, the use of ontologies here is tailored towards inter-agent communication and does not address reasoning over distributed knowledge bases.

3 The Case for an Ontology-based Approach

As motivated in the introduction, there are technical and organizational reasons that demand a decentralized approach to infrastructure monitoring as well as particular requirements on information modelling:

- *Heterogeneous data sources:* Information to be integrated includes information obtained by human observation, deployed sensor systems, and external services. These sources are highly heterogeneous and use their own proprietary data models.

- *Multiple stakeholders*: Multiple organizations usually collect state information about infrastructures independently from each other. Integration is therefore not only hindered by lack of semantic interoperability, but also organizational obstacles.

- *Geographically large scale*: Typical infrastructures such as rail networks and power networks are very large-scale and therefore comprise a large number of geographically distributed components.

- *Long-life system with continuous adaptation and extensions:* Infrastructures inherently have a long life-cycle and undergo continuous change. Information modelling must be sustainable and support extension and maintenance.

- *System must support uses that are not yet known:* In addition, requirements on information modelling are expected to change over time. Therefore, the information model cannot be tailored to a specific set of functionalities, but must remain flexible.

- *Diverse and complex relations between infrastructure elements*: There are complex interdependencies between infrastructure elements, which have to be appropriately captured and represented by the information model.

Based on this analysis, we propose to adopt formal knowledge representation techniques for modelling information in infrastructure monitoring systems. Knowledge representation and reasoning (KRR) is an active field of research in Artificial Intelligence (AI). At its core, this area is concerned with identifying reasonable trade-offs between expressivity of representation and complexity of reasoning. Applied to infrastructure monitoring, KRR carries the following advantages:

- Information is represented with respect to the knowledge model and independent from particular queries. It is therefore "open-ended" and not limited to answering only anticipated queries.

- KRR formalisms usually adopt the so-called Open World Assumption (OWA), which means that the available knowledge is inherently assumed to be incomplete. This is particular useful for decentralized systems with high dynamics where it would be unrealistic to assume complete knowledge.

- Due to their foundation in formal logics, KRR frequently supports explanations of inferences. This is an important feature in infrastructure monitoring, where information from a multitude of sources is integrated in order to derive conclusions.

- Finally, an ontology represents a *declarative* conceptualization of the domain of discourse. This allows humans to quickly understand the available information and, in particular, facilitates maintainability and extensibility.

As KRR formalism for our application domain, we select Description Logics (DL, a family of first-order logic-based languages [BCG03]) due to the following reasons:

- *Recent theoretical advances*: Interesting logics within the Description Logics family have recently been identified, which offer high expressivity together with tractable reasoning complexity. Examples are SHIQ [HST00] and SHOIN [HP03].

- *Implementations and tools*: In addition to these algorithms, mature and highly optimized implementations are available both as open-source (Pellet, SHOIQ) and commercial closed-source (RacerPro, SHIQ)[1]. These are complemented by tools for engineering ontologies such as Protégé and SWOOP[2].

- *Standards and availability of ontologies*: A major driver for ontology-based knowledge representation has definitely been W3C's Semantic Web initiative. The publication of the Web Ontology Language OWL [GH04] as an XML-based standard for representing ontologies enabled interoperability among tools and fostered the publication of ontologies in a common format. Additional thrust can be expected from ongoing standardization efforts such as for query languages (SPARQL [PS07]) and rule representation languages (SWRL [HPB04]).

---

[1] Pellet: http://pellet.owldl.com/, RacerPro: http://www.racer-systems.com/

[2] Protégé: http://protege.stanford.edu/, SWOOP: http://code.google.com/p/swoop/

4 Decentralized System Architecture

This section proposes a functional architecture for a decentralized and ontology-based infrastructure monitoring system. It structures this functionality into layers so that each layer depends only on the next layer down and specifies interfaces for each layer. This way, a high degree of flexibility is achieved as layers can be replaced without affecting the overall architecture. In particular, it also facilitates reuse of components within the same layer due to homogeneous interfaces. We will first describe the different layers and then present different possible deployments.

4.1 Functional Architecture

An important property of the proposed approach is the ontology-based system information model. This model can be arranged orthogonally to the functional layers and provides a common information model, which is shared across all layers. This system information model represents a conceptualization of the domain of discourse, e.g. the rail domain or power network domain. It is important to stress that the system information model is not tailored towards particular data sources, but is a formal representation of relevant knowledge in the application domain. This is also why we adopt the generic term *context information* as used on the field of context-aware computing [De01], which covers any relevant information about the infrastructure including information from external sources. Remaining independent from data sources ensures flexibility for answering arbitrary queries and enables interoperability and future extensibility. The functionality of the system is structured into four layers (see Figure 1):

The ***context acquisition layer*** comprises existing sensor systems and other data sources, which provide context information about the current state of infrastructure parts. The functionality of this layer is to obtain this information from these sources in their highly heterogeneous and proprietary format and to represent it with respect to the ontology-based system information model. As these transformations are source-specific, they will usually be carried out by source-specific wrappers.

The ***context management layer*** comprises repositories, which store and manage the obtained context information. They assume this information in an ontology-based representation from the context acquisition layer. The functionality of this layer is to store, manage and provide access to the acquired context information together with its history. As it handles ontology-based data, the repositories will usually be triple stores, i.e. database management systems that are optimized for handling triple-based data.

The ***reasoning layer*** comprises reasoning components, which provide reasoning services on the information provided by the context management layer. Reasoning services include ontology-based reasoning, which exploits the semantics expressed in the system information model in order to derive implicit information (see also Section 5). They may also include other specialized reasoning algorithms such as topological reasoning services, which process the available information in a different way.

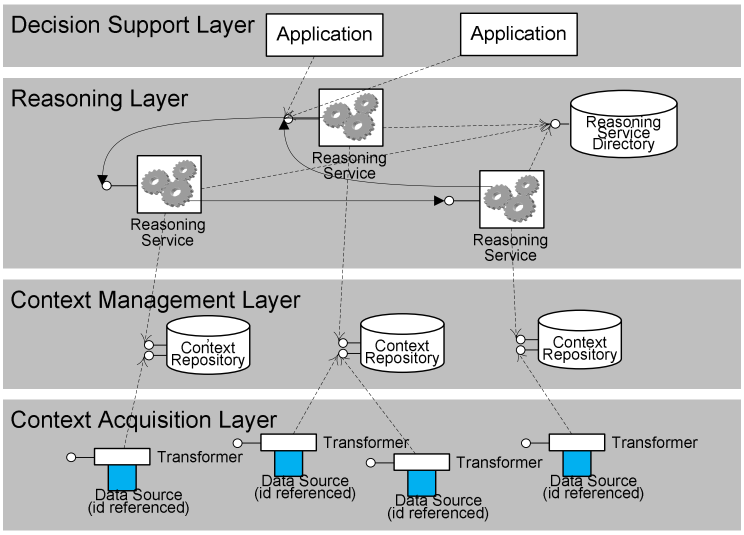

Figure 1: Layers of the Functional Architecture

The ***decision support layer*** comprises application-specific business logic that provides decision support to the infrastructure manager. This includes planning and optimization algorithms, which revert to the reasoning layer in order to take into account up-to-date and historical context information about infrastructure parts.

4.2 Potential Deployments

The previously introduced functional architecture represents a conceptual structure of the components of an infrastructure monitoring system. There are different ways of realizing this architecture in practice and deploying it physically. Here we compare a centralized deployment option with a decentralized deployment option.

Centralized Deployment: Due to the existing and geographically distributed sensor systems, even in the centralized approach, the *context acquisition layer* will be highly decentralized. Transformation from proprietary formats to an ontology-based representation can be centralized by collecting sensor data at a central site and performing the transformation there. The *context management layer* will be realized as a central data store at the system's command-and-control centre collecting all the data. The *reasoning layer* will be co-located with this data store and offer different algorithms for processing and refining the collected data. Finally, the *decision support layer* will offer appropriate optimization and planning algorithms for supporting the command-and-control operator.

Decentralized Deployment: The *context acquisition layer* is given to be highly decentralized. The transformation to an ontology-based representation will take place close to the data sources. The *context management layer* will be highly decentralized with a repository at the site of each operator of a sensor system. Consequently, the *reasoning layer* will also be realized in a decentralized manner with reasoning services located nearby each repository and collaborating for performing reasoning tasks (see Section 5). Finally, the *decision support layer* will also comprise infrastructure manager-specific components, which make use of the reasoning layer in order to support the infrastructure manager.

The centralized deployment carries the advantages of a simple architecture with centralized control. On the other hand, it suffers from limited robustness and scalability as the central components are single points of failures and represent bottlenecks with respect to handling incoming data and queries. With sensor systems continuously generating data, traffic between sensor systems and the central data store can be prohibitively high. The decentralized deployment, in contrast, exhibits particularly high robustness as the failure of a repository or reasoning component does not lead to the failure of the overall system. However, it will lead to the unavailability of certain context information. This is addressed by adopting the open world assumption for the system information model, which assumes knowledge bases to be inherently incomplete. Thus, lack of information results in inference of less, but at least as much as possible information. The same applies to addition of repositories and reasoners: Adding any kind of information (referring to the systems information model) improves and increases the amount of implicit information that can be inferred. This makes the decentralized approach very extensible and scalable. Centralized control is difficult to achieve in this setting. However, the organizational structures in infrastructure monitoring actually imply a decentralized approach.

5 Conjunctive Queries over Distributed Knowledge Bases

We previously argued that decentralized deployments carry essential benefits for infrastructure monitoring systems. Decentralizing information storage, however, is at the cost of increased effort for processing this information. This section therefore focuses on the reasoning layer and presents an approach for realizing distributed ontology-based reasoning. More precisely, we deal with answering grounded conjunctive queries (with respect to the system information model) over distributed knowledge bases.

5.1 Preliminaries

The problem is formalized as follows (see Figure 2 for an illustration): R_i is a *reasoning service* associated with a knowledge base K_i and answers conjunctive queries with respect to K_i. R_i can also submit conjunctive queries to any other reasoning service R_j.

A *knowledge base* consists of a so-called TBox (terminological box) T and an ABox (assertional box) A. The TBox specifies the relevant classes of objects in a domain together with their general relations. It therefore formalizes the semantics of a domain and represents the domain model. The ABox contains assertions about a particular state of the domain, i.e. the objects that are currently available, their properties and actual relations. Measurements and events from sensor systems are therefore represented as instances in the ABox. Note that in our case, T is common to all K_i, but A is split into separate A_i with $A = \cup_i A_i$. So every R_i is associated with a knowledge base $K_i = (T, A_i)$. We also assume that for every ABox instance a, which is mentioned in more than one ABox A_i, it holds that a is asserted to be a member of only the special concept C_{shared} with C_{shared} disjoint from all other concepts and not used in the definition of other concepts. This ensures that the global knowledge base does not become inconsistent without being detected, and ensures sound and complete answers of the presented query answering algorithm. In the rail domain, C_{shared} could for example be *TrackSegment*, which serves as a shared reference for all sensor systems.

Also associated with each reasoning service is *Profile*(R_i), which represents an abstract of the knowledge that can be provided by R_i as a means for optimizing query answering. Finally, there is a *directory service D*, which indexes *Profile*(R_i) for all i.

The query to be answered is a *grounded conjunctive query* $Q = q_1 \wedge \ldots \wedge q_n$. q_i can be a *concept query atom* of form $C(x)$ or a *property query atom* of form $p(x,y)$, where C refers to a concept and p refers to a property in T, and x, y are either variables or refer to instances in A. Answers to Q are all possible bindings of named variables to instances in A so that the imaginary global knowledge base $K = (T, A)$ entails all q_i.

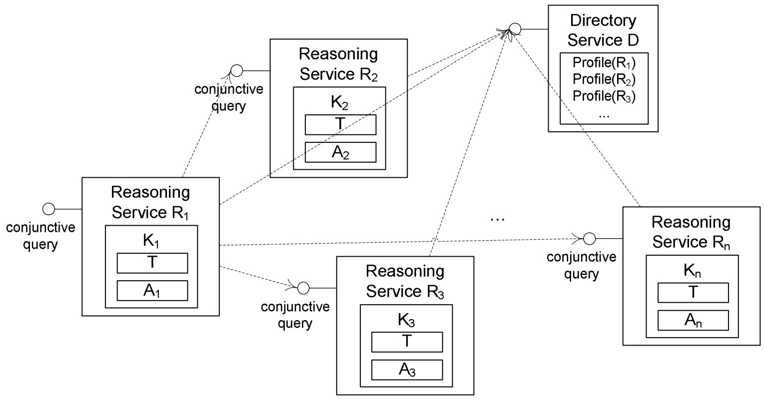

Figure 2: Formalization of the Distributed Reasoning Problem

5.2 Answering Queries

The basic idea is to answer conjunctive queries by generating subqueries to the local knowledge base as well as remote reasoning services and then integrating the subanswers to correct complete answers. The initial query can be posed to any reasoning service. It then analyzes the query, generates a query execution plan, executes the plan, and returns the consolidated answers.

The **query execution plan** is constructed conjunct by conjunct: For each conjunct, the relevant reasoning services, which can contribute to the overall answer, are identified. Conjuncts assigned to the same reasoning service are subsequently clustered.

Identification of relevant reasoning services is done through the directory service, which manages the profiles of all known and available reasoning services. Note that this directory service can either be realized as a centralized registry or again in a decentralized manner in order to guarantee robustness.

Profile(R_i) of a reasoning service R_i is defined as a set of concept and property names from T:

- Concept name C is element of *Profile*(R_i) iff there is at least one instance a mentioned in A_i (associated with R_i) which can be inferred to be an instance of C, i.e. K_i entails $C(a)$.
- Property name p is element of *Profile*(R_i) iff it can be inferred for at least one pair of instances a, b in A_i (associated with R_i) that p holds, i.e. K_i entails $p(a, b)$.

In both cases, only the *most specific concept name* and the *most specific property name* [BCG03], respectively, are included for exploiting concept and property hierarchies when determining the relevance of R_i for q_i: For each R_i, check whether *Profile*(R_i) contains a concept name C/property name p, which is *equivalent to or subsumed by* the concept name/property name used in q_j (i.e. T entails that C is subsumed by C_{qi} or T entails that p is subsumed by p_{qi}). If and only if this is the case, then R_i is considered relevant for answering q_j.

Profiles need to be up-to-date in order to ensure up-to-date answers. As each modification of knowledge base K_i could imply a change to *Profile*(R_i) and knowledge bases are permanently updated with new sensor information, an appropriate design of profiles is necessary in order to reduce the number of profile updates. Therefore only concept and property names from T are included and instance names from A are omitted: We assume that knowledge bases usually manage ABox data of only a subset of T (e.g. only information about wheel impact load measurements) and that different knowledge bases usually manage data of rather disjoint subsets of T (e.g. one manages wheel impact load data and the other manages hot axle box data). Based on these assumptions, constructing profiles only with respect to T results in few profile updates.

The result is an initial query execution plan that associates query conjuncts with a set of relevant reasoning services.

Query optimization is performed by adapting the order of query conjuncts in the query execution plan. This step uses information on selectivity of query conjuncts in order to adjust the order of query conjuncts so that the most specific conjuncts, which produce only few potential bindings for variables, will be executed first. Evaluation of subsequent query conjuncts can than be limited to these few potential bindings.

Query execution is finally done using an extensible query execution engine based on iterators [Gr93]. If more than one reasoning service is relevant for the same query conjunct, answering this conjunct is parallelized and the results are subsequently unified with duplicates being eliminated.

6 Prototype and Evaluation

As an important design decision, we realize a reasoning service by combining an existing reasoner implementation with a custom execution engine for distributed queries. The query engine is responsible for constructing query plans and integrating bindings from different sources. The reasoner implementation, on the other hand, is considered a black-box component, which can be initialized with a knowledge base and subsequently answers queries with respect to this knowledge base. That means that (1) no modification to existing reasoner implementations is necessary and (2) reasoning functionality is encapsulated so that we can easily switch to a different (e.g. more powerful or efficient) reasoner implementation in the future.

The current **prototype implementation** encompasses a reasoning service, which is instantiated multiple times, and a directory service. The reasoning service provides a HTTP interface for accepting external SPARQL queries and returns answers as an RDF document. It consists of a custom query engine, which implements the previously described query answering algorithm, and is based on the ARQ query engine of HP's Jena Semantic Web Framework[3]. As reasoning component we adopted the Pellet reasoner[1]. The directory service also uses Jena and provides a HTTP interface for managing profiles as well as requesting relevant reasoning services for a given query conjunct.

Conjunctive queries are accepted in the SPARQL language. The mapping of concept and property query atoms to triples in the SPARQL-WHERE clause is straightforward:

$C(x)$ is mapped to {?x rdf:type C.}, $p(x, y)$ is mapped to {?x p ?y.}

Using the prototype implementation, we ran **first evaluation experiments** for comparing query answering performance of the centralized with the decentralized approach. As an example data set, we use a custom ontology from the rail infrastructure domain. Among others, this ontology covers rail profile, track geometry, and infrastructure network. Rail profile is concerned with wear of a single rail, while track geometry deals with the correct alignment and gauge of two rails. There are sensor systems in place, which generate data for both aspects. Infrastructure network defines the concepts necessary for describing a particular rail infrastructure. The test query involves all types of data: It asks for all track segments connected to a given track segment, which are inferred – based on the sensor system events collected at the different sites – to have both critical rail profile and track geometry states.

The experiments with four reasoning services and one directory service were run on one Core Duo 1.83GHz machine with 1GB RAM. R_1 receives the initial query, R_2 is responsible for the infrastructure network, R_3 for rail profile data, and R_4 for track geometry data. We ran the described query on different sets of rail profile-related, track geometry-related and infrastructure network ABox data. The datasets on rail profile and track geometry-related data were increased in size. For each ABox configuration, the

---

[3] Jena Semantic Web Framework: http://jena.sourceforge.net/

query was evaluated both centrally and distributedly. Central evaluation first collected all remote data and then performed the reasoning locally, while distributed evaluation split up the queries as described before.

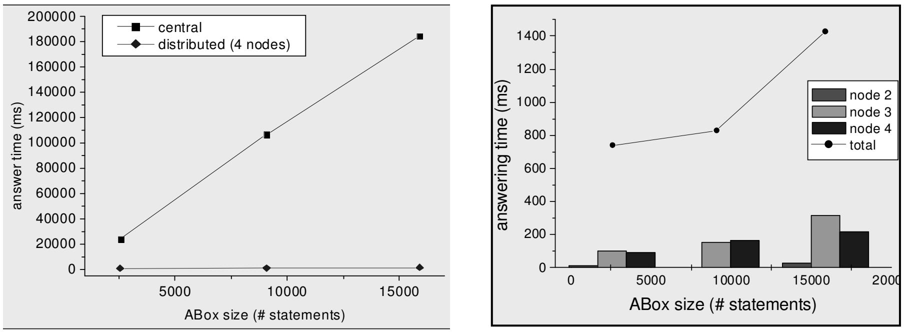

Figure 3: Answer time for centralized and distributed query evaluation for increasing size of knowledge bases (left), break-down of remote reasoning times in distributed case (right)

Both centralized and distributed evaluation returned the same answers. Figure 3 shows first results on query answering performance: On the left, answer time of central evaluation is compared with answer time of distributed evaluation. As expected, distributed evaluation is much faster, as data traffic is limited to (potential) answers and reasoning load is balanced across reasoning services. The performance gain increases with increasing knowledge base size. On the right, reasoning load on the different reasoning services (for the distributed evaluation) is compared. For R_2 it remains constant as R_2's knowledge base does not increase in size. R_3 and R_4, however, show increasing load with increasing knowledge base size. It can also be seen that data traffic still accounts for a significant amount of overall answer time.

7 Conclusion

This paper introduced physical infrastructure monitoring as an application domain that demands decentralized system designs. Reasons for this are the large scale of these systems combined with high robustness requirements, the heterogeneity of data sources as well as the multitude of stakeholders involved. In order to tackle the semantic interoperability problem, aggravated by adopting a decentralized approach, we proposed to adopt Description Logics-based ontologies. As a foundation for designing ontology-based infrastructure monitoring systems, we developed a layered functional architecture and illustrated potential deployments. In particular, we detailed decentralization of the reasoning layer and described systems design, query answering algorithms and a prototype implementation. Future work is focused on advancing query answering to more powerful semantics and conducting detailed performance evaluations.

References

[BCG03] Baader, F., Calvanese, D., McGuinness et al.: The Description Logic Handbook. Cambridge University Press, 2003.

[De01] Dey, A.: Understanding and using context. In.: Personal and Ubiquitous Computing, 5(1): 4-7, 2001.

[BPR06] Botts, M., Percival, G., Reed, C., et al: OGC Sensor Web Enablement: Overview and High Level Architecture. OGC 06-046r2, 2006.

[ERR02] The European Rail Research Advisory Council (ERRAC), Strategic Rail Research Agenda, 2002.

[FSW05] Feng, J., Smith, J., Wu, Q. et al.: Condition Assessment of Power System Apparatuses Using Ontology Systems. In: Proc. of Transm. and Distr. Conf., IEEE, Dalin, 2005.

[Gr93] Graefe, G.: Query evaluation techniques for large databases. In: ACM Computer Surveys, 25(2): 73-170, 1993.

[GH04] McGuiness, D., van Harmelen, F.: OWL Web Ontology Language Overview. W3C, http://www.w3.org/TR/owl-features/, 2004.

[HST00] Horrocks, I., Sattler, U., Tobies, S. et al.: Reasoning with Individuals for the Description Logic SHIQ. In: Proc. of 17th Int. Conf on Automated Deduction, Springer, 2000.

[HP03] Horrocks, I., Patel-Schneider, P.: Reducing OWL entailment to description logic satisfiability. In: Proc. of 2nd Int. Semantic Web Conference, Springer, 2003.

[HPB04] Horrocks, I., Patel-Schneider, P., Boley, H. et al.: SWRL: A Semantic Web Rule Language. W3C, http://www.w3.org/Submission/SWRL/, 2004.

[KRL05] Kezunovic, M., Ren, Z., Latisko G. et al.: Automated Monitoring and Analysis of Circuit Breaker Operation. IEEE Trans. on Power Delivery, 20(3), pp 1910-1918, July 2005.

[KT04] Kezunovic, M., Taylor, H.: New solutions for substation sensing, signal processing and decision making. In: Proc. of 37th Hawaii Int. Conf. on System Sciences, 2004.

[Lag07] Lagnebäck, R.: Evaluation of wayside condition monitoring technologies for condition-based maintenance of railway vehicles. PhD thesis, Lule Univ. of Technology, 2007.

[MRS05] Maly, T., Rumpler, M., Schweinzer, H. et al.: New development of an overall train inspection system for increased operational safety. In: Proc. of Intelligent Transportation Systems, IEEE, 2005.

[MS06] Moodley, D., Simonis, I.: A New Architecture for the Sensor Web: The SWAP Framework. In: Proc. of Int'l Semantic Web Conf., Semantic Sensor Networks, 2006.

[PS07] Prud'hommeaux, E., Seaborne, A.: SPARQL Query Language for RDF. W3C, http://www.w3.org/TR/rdf-sparql-query/ , 2007.

[RB01] Ruback, L., Balke, K.: Integrating Train Information for Advanced Transportation Management. Texas Transport. Institute, National Technical Information Service, 2001.

[RFP06] Lewis, R., Fuchs, F., Pirker, M. et al.: Using Ontology to Integrate Railway Condition Monitoring Data. Int'l Conf. on Railway Condition Monitoring, Birmingham, 2006.

[SLW07] Schröder, A., Laresgoiti, I., Werlen, K. et al.: Intelligent Self-Describing Power Grids. In: Proc. 19th Int. Conf. on Electricity Distribution, Vienna, 2007.

[ST98] Steets, P., Tse, Y.: Conrail's integrated automated wayside inspection. Proc. of the 1998 ASME/IEEE Joint Railroad Conference, 1998.

[TK07] Terziyan, V., Katasonov, A.: Global Understanding Environment: Applying Semantic Web to Industrial Automation. In: Emerging Topics and Technologies in Information Systems, IGI Global, 2007.