Knowledge-Based Self-Organization of Traffic Control Systems

Matthias Jurisch[1] and Bodo Igler[1]

Abstract: Traffic control systems operating at the level of intersections can interact with each other. This interaction can be implicit (traffic flow) and explicit (exchange of sensor data). A central issue in this context is how to react to structural changes in a system of traffic control systems. This paper proposes to model all aspects which are relevant to the connection of these systems as ontologies. It further proposes to adapt to structural changes by taking inferences drawn from these ontologies into account. This work in progress is presented with the help of a concrete application example. A software prototype has been developed to demonstrate that this approach is technically feasible.

Keywords: Self-Organization; Ontology; Traffic-Control

1 Introduction

In increasingly complex, heterogeneous information systems, automating integration is a very important task. Systems that have to work together but have not been designed to cooperate at runtime are called *interweaving systems* [To16]. These systems need to behave consistently with their environment which consists of other systems and the communication with them. We consider the exchange of data and physical interactions as the communication between the systems. Therefore, a central problem of this field is how changes should be addressed. A model of the environment can be expressed using ontologies based on description logics. Using ontologies in this context allows developers to benefit from a rich set of existing standard tools that are easy to use but nevertheless provide a sufficient expressiveness. In this paper, we propose an ontology-based approach for modeling system environments and show how the model can be adapted to changes. From this information, each system needs to construct its own model of the environment. As an application example for interweaving systems we use autonomic traffic control at the level of intersections. Each system represents one intersection.

Our approach is applied to manage a model that is used to select traffic data from related junctions. This data can be used to classify traffic situations and adapt signaling plans based on a set of rules and information observable in traffic data. When systems at other junctions change (e.g., they are deactivated or fail, are reactivated, or underlying structures

[1] Department of Design - Computer Science - Media, RheinMain University of Applied Sciences, Unter den Eichen 5, 65195 Wiesbaden, matthias.jurisch@hs-rm.de, bodo.igler@hs-rm.de

doi:10.18420/in2017_97

change because of roadblocks, congestion, or other traffic conditions), the model needs to be adapted to find e.g., replacements for missing data sources.

Similar problems have been addressed in approaches to the mapping adaption problem [VMP03]. This problem deals with adapting mappings between data models to changes. We tackle this issue by applying an inference-based difference computation approach [Ju16]. Our approach is able to correctly classify data measurements from other junctions and to adapt the environment model of each traffic control system to these changes. The problem lying at the core of our approach is also relevant to various areas regarding self-management and the engineering of distributed technical systems. Since these systems get more complex with technological advances, it is especially important to make approaches easily maintainable. Ideally, most of the work should be conducted using standard modeling technologies and programming should be used as little as possible [Se03].

The remainder of this paper is structured as follows: Section 2 discusses related approaches from the literature and identifies the core research questions of this work. The application example used to demonstrate our approach is presented in Section 3. The approach itself is presented in Section 4. This section discusses how the model of the application example is structured (Section 4.1), how differences between different versions of models are computed (Section 4.2) and how the approach is implemented (Section 4.3). Advantages and disadvantages of the approach are discussed in Section 5. Section 6 gives a conclusion.

2 Foundations and Related Work

Current approaches for autonomic traffic control, e.g. [STR16], typically consider intersections that do not exchange traffic data or other information to improve each other's traffic management performance. Using information about traffic congestions or other traffic conditions at related intersections can be useful, especially in situations where phased traffic lights are used. [Pr09] presents a peer-to-peer synchronization mechanism that allows intersection-based traffic control systems to agree on a common signaling phase. However, these systems do not consider structural changes. While ontologies have been used in the field of traffic control, their usage focuses on providing a uniform view on the sensor data of traffic control systems. E.g., [Fe16] proposes an ontology-driven approach to traffic control systems that uses an ontology layer as a standardized representation of traffic situations. However, it is still unclear how structural changes should be addressed in the context of these systems. They might receive information that contradicts or fundamentally changes the local view, i.e., the internal knowledge of the system used to interpret traffic data and adapt the signaling plans. In summary, the remaining issue is how to address structural changes in this application domain.

One possibility to address this issue is to base the approach on ontology mapping adaption. Problems similar to mapping adaption have been studied in the context of database and schema mapping research. An incremental approach was proposed by [VMP03]. It reacts to changes based on rules defined via change patterns. Each pattern defines a corresponding modification for the mapping. Another approach is based on the composition of mappings [YP05]: A new mapping between one model a and an evolved version b' of a model b, where a mapping between a and b already exists, is created by a composition of the mapping between a and b and a mapping between b and b'. [Gr13] have applied these ideas to ontology mappings. While inferences could be considered when working with ontologies, the impact on the inferences of ontologies on the mapping adaption is not taken into account by this approach. A recent survey on this subject has concluded that current approaches do not consider formal axiom-based-ontologies but focus on simple *is-a* and *part-of* hierarchies [GPR16]. A survey of the state of the art in mapping adaption comes to similar conclusions and identifies the question of how to consider the semantics of changes and reuse knowledge from old mappings [DPRD15]. To consider the semantics of ontologies, the inferences that can be drawn from them need to be taken into account. The issue at the core of our research is how inferences can be included into the mapping adaption process and how this idea can be applied to adapting environment models of traffic control systems to structural changes.

3 Application Example

We will demonstrate our approach with the help of an interweaving system example: A traffic control system managing an intersection uses traffic measurements to optimize its signaling plan. The changes we consider are (1) addition and removal of traffic control systems and (2) traffic congestions or road blocks. These changes are considered by each individual system on the intersection level.

In our model, at each intersection the traffic flow in and out of the intersection is measured for every road entering the intersection. To optimize the traffic flow management process these measurements are exchanged with systems at other intersections performing the same task. To reduce the effort needed to design the system and make it more scalable, the traffic control systems themselves find out what data from remote intersection is relevant to them. Data that is determined to be relevant is included in the traffic management process.

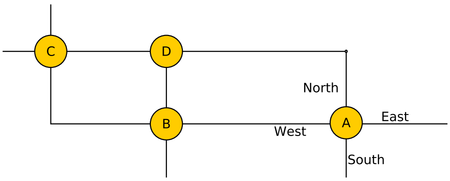

Fig. 1: Map of the application example

A map of the intersections used as application example in this paper is depicted in Figure 1. The junctions *A, B, C* and *D* are depicted as yellow circles, roads are shown as black lines. Lines not connected to intersections lead to areas outside of the scope of our example. The identifiers of roads leading to the intersections are exemplarily depicted for intersection

A. The traffic management systems exchange sensor data. Aggregated sensor values of all intersections are compared and a correlation score is computed. It is also possible to only consider the closest intersections to make the computation of correlation scores scalable. Sensor readings with the highest correlation score are used to make decisions on signaling plans. When a system's model of available data is complete, it can locally apply its traffic management strategy.

4 Approach

To adapt the internal model of each traffic management systems, we apply an approach for easy integration and data selection. To reach this goal, we use three models. A global model represents the only shared global information and contains only schematic knowledge about available data. A correlation model contains statistical information about the connections between intersections. A local environment model contains mappings between local requirements and remote data sources. When changes occur, rules based on the aforementioned correlation scores and connections define how to adapt the mapping. The changes are evaluated on differences of inferences that can be drawn from the model.

4.1 Modeling

To represent the environments of the intersections, each intersection has access to three ontologies: (1) a global ontology that contains schematic knowledge about the data provided by all intersections, (2) a local correlation ontology that holds information on how the local system is connected to other intersections (3) a local ontology of each intersection that connects local data requirements to data providers from remote intersections via an ontology mapping. In this way, the global ontology serves as a kind of interface format for the data provided by each intersection. To allow reasoning in resource-constrained settings, the models heavily rely on subset relations that even primitive reasoners can process efficiently.

A graffoo diagram [Fa14] of the local ontology for intersection A including a mapping to the global ontology is depicted in Figure 2. In this type of diagram, classes are represented as rectangles, data types are depicted as rhombuses and instances are represented by circles. Classes that are part of the global view are depicted in light blue. Every *Intersection* instance can be related to a timestamp and a sensor reading. All data that is relevant to traffic planning decisions is stored as an instance of this class. From a global perspective, there is one direct subclass[2] for each intersection. The classes represented in this diagram (*A, B, C, D*) correspond to the intersections from the application example. Intersection classes have one direct subclass for each entry into the junction (e.g., *BEastIn, DWestIn*). This modeling approach does not rely on the specifics of the application example, since intersection classes and subclasses for junction entries can be automatically generated in the general case.

[2] The semantics of subclass relations is used to express subset relation in this paper.

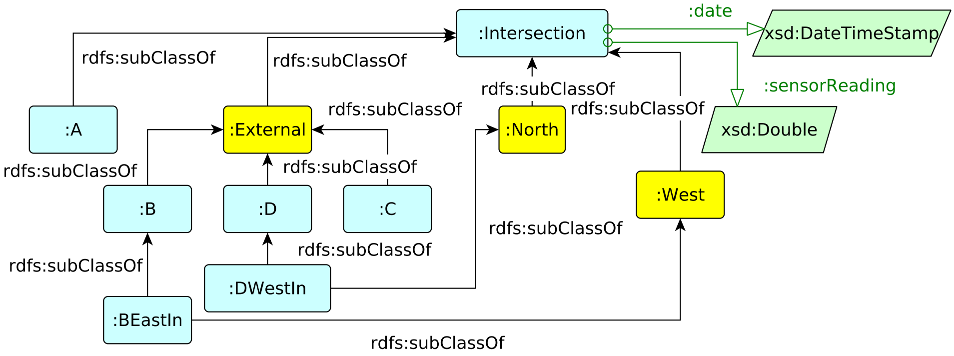

Fig. 2: Conceptual model of local ontology *A* including mapped entities from the global view

The local ontology classes for intersection *A* are shown in yellow. Each intersection needs data from remote intersections. The local ontology is used to connect local needs for data from remote junctions to the respective data providers. This view also specifies which parts of the model are external, i.e., not directly measured at the current intersection. Hence, in the presented example *B, C* and *D* are subclasses of *External*, while *A* is not. To express that an external data source is needed, the classes *North* and *West* are used. To express that a need for data is fulfilled by a data source, a subclass relation is stated (e.g., data source *BEastIn* is a subclass of *West*).

To determine which of these local classes should be connected to which global classes, information regarding statistical traffic correlations is used. We call this information background knowledge, since it provides additional information regarding the structure of the application example. The details for the computation of this correlation are out of the scope of this paper. We therefore take in our example the correlation values as given. A graffoo diagram of the conceptual model of this background knowledge at intersection *A* is shown in Figure 3. Correlation data entities are shown in pink, items from the local model in yellow and global parts are shown in blue. The correlation between two roads entering intersections is represented by the class *CorrelationEntry*. This class is related to the classes *LocalConnection*, which represents the local driveway and the class *NamedConnection*, which represents the remote junction data. Figure 3 contains an example for a correlation between *A West* and *B East*, represented by the instance *WestBEastIn*. This type of correlation exists for all pairs of roads leading to intersections and determines which connections shown in Figure 2 are established, i.e., the correlation data is used to adapt the local ontologies to changes.

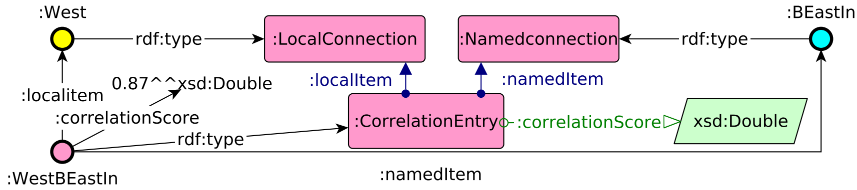

Fig. 3: Conceptual Model of Correlation Data

4.2 Adapting to Changes

Since connections between local and global data are modeled through ontology mappings, an adaption of the local ontologies is required when changes occur. To adapt to changes, each traffic control system executes an algorithm based on an approach described in [Ju16]: When changes in the environment occur, each system has to evaluate the semantics of these changes. Semantics of the knowledge base k are defined as the set of inferences $\inf(k)$ that can be drawn from the model using standard ontology engineering tools. It is possible to compute differences of the inferences $\Delta_{\inf}$ of the knowledge base k before ($\inf(k)$) and after the change ($\inf(k')$). One example for a relevant change is the deletion of a subclass of *External*, which would occur in the event of the removal of a traffic management system. Replacement candidates are found based on the correlation entries described in Section 4.1.

When differences are found, $\Delta_{\inf}$ is searched for entries that are relevant to instances of the class *External*. These changes indicate which connections need an update. The best candidates for an update are found using the following SPARQL graph pattern that is executed on the correlation data ontology described in Section 4.1:

```
SELECT ?dataClass  ?score WHERE {
  ?corrItem :namedItem ?dataClass;
  :localItem <"+connectionNeedsNew.getURI+">;
  :correlationScore ?score
} ORDER BY DESC(?score)
```

In this query, `connectionNeedsNew.getURI` is replaced by the URI of the connection that needs an update. The results are all data classes that have a correlation with the connection that needs a new data input, ordered by correlation score. The results of this query are then checked for consistency and the mapping statements in the local ontology are updated.

4.3 Implementation

To demonstrate that the presented approach can work, we implemented a prototype. All models are represented as OWL ontologies [Hi09] and accessed using the Jena API [Ap17].

The prototype reads in two versions of a local ontology according to the example described in Section 3. The implementation contains a domain-independent framework that computes the inference differences Δ_{inf} using the internal Jena Reasoner and provides hooks for reaction to changes. Using these hooks in the domain-specific part of the prototype, reactions are based on graph patterns formulated in SPARQL to find the best candidates for mappings depending on occurred changes and background knowledge. To test the implementation, we used two structural change scenarios for the described application example. The first scenario is a removal of an intersection, the second an addition. The prototype was able to propose the correct changes to the mapping in each scenario.

5 Discussion

The domain-specific aspects of the approach were implemented with very few lines of source code (less than 100 lines of scala code). The remainder of the domain-specific implementation was accomplished by formulating ontologies using standard semantic-web tools. This is a major advantage of this approach. Having more parts of the approach specified in a standardized format with explicit semantics means fewer costs for adapting the software to changes in the environment. This also means that fewer code changes are required to adapt the approach to other application domains.

To implement the proposed process, no domain-specific modeling tools need to be created. Engineers can use freely available tools that have been proven to work in the ontology engineering community. When problems occur, engineers can turn to this community for help. The way the approach is structured, developers can focus on invariants and rules that describe how the systems should react to changes, which facilitates a clear separation of structural and behavioural aspects.

6 Conclusion

In this paper, we presented an approach to the management of ontology mappings and demonstrated how it can be used to manage structural changes in autonomic traffic control. We have shown that this process can be implemented mainly using semantic web technologies. The main advantages of this approach are the conceptual separation of different aspects and the reliance on standard modeling languages. As future work building on these ideas we suggest a quantitative evaluation of the approach and its application to traffic control systems. Furthermore, we consider an application to other areas, like maintaining mappings in software engineering repositories or other use cases for interweaving systems.

References

[Ap17] Apache Foundation: Apache Jena. 2017. `https://jena.apache.org/`, website [retrieved: May, 2017].

[DPRD15] Dos Reis, Julio Cesar; Pruski, Cédric; Reynaud-Delaître, Chantal: State-of-the-art on Mapping Maintenance and Challenges Towards a Fully Automatic Approach. Expert Systems with Applications, 42(3):1465–1478, 2015.

[Fa14] Falco, Riccardo; Gangemi, Aldo; Peroni, Silvio; Shotton, David; Vitali, Fabio: Modelling OWL ontologies with Graffoo. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8798:320–325, 2014.

[Fe16] Fernandez, Susel; Hadfi, Rafik; Ito, Takayuki; Marsa-Maestre, Ivan; Velasco, Juan: Ontology-Based Architecture for Intelligent Transportation Systems Using a Traffic Sensor Network. Sensors, 16(8):1287, 2016.

[GPR16] Groß, Anika; Pruski, Cédric; Rahm, Erhard: Evolution of Biomedical Ontologies and Mappings: Overview of Recent Approaches. Computational and Structural Biotechnology Journal, pp. 1–8, 2016.

[Gr13] Groß, Anika; Dos Reis, Julio Cesar; Hartung, Michael; Pruski, Cédric; Rahm, Erhard: Semi-automatic Adaptation of Mappings between Life Science Ontologies. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7970 LNBI:90–104, 2013.

[Hi09] Hitzler, Pascal; Krötzsch, Markus; Parsia, Bijan; Patel-Schneider, Peter F.; Rudolph, Sebastian, eds. OWL 2 Web Ontology Language: Primer. W3C Recommendation, 27 October 2009. Available at `http://www.w3.org/TR/owl2-primer/`.

[Ju16] Jurisch, Matthias: Managing Ontology Mapping Change based on Changing Inference Sets. In: Knowledge Engineering and Knowledge Management - EKAW 2016 Satellite Events, EKM and Drift-a-LOD. Bologna, Italy, November 19-23, 2016. Revised Selected Papers. Springer, pp. 255–262, November 2016.

[Pr09] Prothmann, Holger; Branke, Jürgen; Schmeck, Hartmut; Tomforde, Sven; Rochner, Fabian; Hahner, J.; Muller-Schloer, C.: Organic Traffic Light Control for Urban Road Networks. International Journal of Autonomous and Adaptive Communications Systems, 2(3):203–225, 2009.

[Se03] Selic, Bran: The pragmatics of model-driven development. IEEE software, 20(5):19–25, 2003.

[STR16] Stein, Anthony; Tomforde, Sven; Rauh, Dominik: Dealing with Unforeseen Situations in the Context of Self-Adaptive Urban Traffic Control: How to Bridge the Gap. In: IEEE International Conference on Autonomic Computing. 2016.

[To16] Tomforde, S.; Rudolph, S.; Bellman, K.; Würtz, R.: An Organic Computing Perspective on Self-Improving System Interweaving at Runtime. In: 2016 IEEE International Conference on Autonomic Computing (ICAC). pp. 276–284, July 2016.

[VMP03] Velegrakis, Yannis; Miller, Renée J.; Popa, Lucian: Mapping Adaptation under Evolving Schemas. VLDB '03 Proceedings of the 29th international conference on very large data bases, pp. 584–595, 2003.

[YP05] Yu, Cong; Popa, Lucian: Semantic Adaptation of Schema Mappings when Schemas Evolve. Proceedings of the 31st international conference on Very large data bases, pp. 1006 – 1017, 2005.