A Knowledge Base for Electric Vehicles in Inner-City Logistics

Thomas M. Prinz, Johannes Kretzschmar, Paul Hempel, and Volkmar Schau

Chair of Software Technology
Friedrich Schiller University Jena, Germany
email: {Thomas.Prinz, Johannes.Kretzschmar, Paul.Hempel, Volkmar.Schau}@uni-jena.de

***Abstract*—Logistics companies depend on the new technology of electric vehicles when inner-city low emissions zones and their restrictions grow. The comprehensible utilization of electric vehicles in such a time and resource critical domain however requires an extensive software support regarding e-vehicle features. Since there are several logistics software systems, there is the need for a knowledge base for electric vehicles to allow a cross-application implementation of those features. In this paper, we argue for such a knowledge base and how it could basically look like. Furthermore, we motivate this base with some use cases. At the end, the paper closes with an exemplary knowledge base for the inference of possible drivers for a specific vehicle type.**

Keywords–Knowledge base; electric vehicles; logistics.

I. INTRODUCTION

The introduction of electric vehicles presents challenges for everyday life since they rise complete new technologies and handling. Otherwise, that introduction becomes more and more important as most big cities. Especially Germany and the Netherlands have low emissions zones restricting the type of vehicles. Inner-city logistics companies depend on that new technology to allow a supply in future as most conventional logistics vehicles have high emissions.

In our research project *Smart City Logistik Erfurt* (SCL) [1], we consider those challenges for the introduction of electric vehicles in inner-city logistics exemplary on the area of the city Erfurt. The major tasks are (1) a range forecast, (2) the driver's acceptance, (3) an open system architecture [2], and (4) a knowledge base:

(1) The range prediction is necessary to enable a precise tour planning since most tours in inner-city logistics should be planned in such that a vehicle uses its full range. Especially in the field of e-mobility, a solid capacity is required to prevent batteries from damage.

(2) The driver's acceptance is important as first tests have shown that the new technology, for example the range restriction of electric vehicles, makes drivers insecure. As result, a system has to support the drivers to school their handling to get a better time/costs ratio.

(3) Since the field of transport management systems offers less open application interfaces, there is the need for building an open system architecture to connect new systems for the consideration of electric vehicles (e.g., a range prediction) to current transport management systems. Before the introduction of electric vehicles can be successful in inner-city logistics, that interaction has to be implemented.

(4) Eventually, the knowledge base comprises necessary information of and behaviour rules for electric vehicles and inner-city logistics. Since there are currently a lot of software systems for logistics, such a base enables a cross-system implementation by system-independent terminologies, interdependencies, and inferences. It is therefore the base for all other mentioned topics and the content of this paper.

Traditionally, the development of knowledge-based systems consists of six steps (c.f. Figure 1): (1) identification, (2) conceptualization, (3) formalization, (4) implementation, (5) testing, and (6) revision [3]. Since there are diverse interdependencies between range influencing factors, we have to perform a *knowledge acquisition* as part of the identification step. Knowledge acquisition in the field of electric vehicles and inner-city logistics requires an analysis of the range influencing parameters, the business processes of logistics companies, and the participants as well as the resources in inner-city logistics, e.g., the structure of delivery tours.

For this purpose, we have to use several knowledge representations in our striven knowledge base. Detailed and structural descriptions of each resource, object, participant in inner-city logistics and in electric vehicles form the foundation. These descriptions define a controlled vocabulary [4] and follow a data-driven system approach [5]. Descriptions of numeric values specify units and their interdependencies, i.e., they allow for an automatic transformation from a source unit into a target unit. The structure of composed terms can be described with groups (compositions), cardinalities (arrays), optionalities, and polymorphisms [5]. Semantic annotations like synonyms, acronyms, textual information, and keywords allow for targeted searches and later comprehensive domain modelling.

Based on that *structural* layer of information, the next layer contains the interdependencies between the different information (or *concepts* in terms of ontologies). Those interdependencies define semantic information which allow for the inference of new or not explicit described information. For example, such a system can infer that a s-pedelec is subsumed by the concept of a moped.

Such a *conceptual* layer builds an advanced ontology for electric vehicles and inner-city logistics. However, the nature of description logics ontologies does not simply and efficient involve numerical interdependencies being the normal case in this field of research. If we consider the classes of European driving licences for example, then we see that one can receive only the driving licence class *B* if that person is at least 18 years old. To represent such a rule, there is the need for an additional layer — a *rule* base. The rule base includes the rules given by the ontology and additional numerical rules.

 ISBN: 978-1-61208-438-1

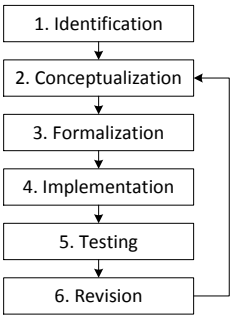

Figure 1. Expert system development after Buchanan et al. [3]

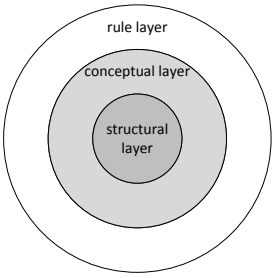

Figure 2. Approach for a knowledge base for electric vehicles

Our overall approach for a complete knowledge base for electric vehicles in inner-city logistics is illustrated in Figure 2. It contains the three above mentioned layers in concentric circles.

In this paper, we motivate some use cases for a knowledge base for electric vehicles in inner-city logistics at first (cf. Section II) and, afterwards in Section III, we consider the use case in European driving licence classes in more detail and demonstrate how we can use our three layer model to describe it. Finally, we close our paper with a short outlook into future work in Section IV.

II. Use Cases

The motivation for the construction of a knowledge base for electric vehicles in inner-city logistics is each of the following use cases, which arose from the SCL project: (1) Infer missing measurement data, (2) infer company important information, e.g., valid tours, drivers which are allowed to drive a specific car, drivers whose driving licence class expires, tour stops in valid time intervals, distances which are feasible for an electric vehicle of the company, checks of driver's rest periods, or goods with the same (or a close) destination and delivery time interval. Furthermore, a knowledge base may (3) provide mechanisms for actual and consistent data, e.g., current available electric vehicles.

In the remainder of this section, we consider these use cases in more detail with regard to their need and an idea for their solution with a knowledge base.

(1). Logistics software products and especially the range prediction use measurement data like global positions (GPS), current speeds, etc. for monitoring and optimization. However, the size of measurement data should be as small as possible without the loss of information. Sometimes, the system has to handle incomplete or contradictory data. For this reason, it (i.e., the knowledge base) has to be able inferring and evaluating the missing data. This is possible for data with physical correlations. For example, the average speed between two measurement points (i.e., two successive received measurement data) can be derived if the GPS positions and time stamps of both measurement points are given. Naturally, in some cases, there is a little loss of quality in the data since, for example, the distance which can be calculated with the help of two GPS positions may vary from the real distance.

Such an inference of measurement data is possible with the help of formal data descriptions, especially the physical correlations, which are defined in the structural layer of our proposed knowledge base.

(2). Logistics companies have the same trend to temporary workers, internationalization, globalisation, and optimization as other companies. For this reason, such companies are confronted with a wide heterogeneity of laws, structures, and cultural characteristics of different countries. It is necessary to collect all these (important) information to allow answers for simple questions whose inference is complex. As mentioned before, such questions could be: What is a good tour that is valid for a specific vehicle and fits all orders? Who of the drivers can drive that vehicle? Is it possible that a driver reaches each tour stop within a valid delivery time interval? When does a driver has to refresh its driving licence class to be continuously usable? Does all drivers observe the legal rest periods? Etc.

For such complex inferences, the structural layer must describe all concepts which belongs to driver licences, drivers, tours, goods, customers, vehicles, street maps, etc. Furthermore, the conceptual layer has to describe the relationships between those concepts and, eventually, the rule layer defines additional rules for those relationships. How such a knowledge base could be implemented is shown exemplary in the next section on the question "Who of the drivers can drive that vehicle?".

(3). In this fast-moving time, it is important to keep up-to-date. If the knowledge base uses standardized data descriptions of concepts, it can automatically support eventual update processes, by checking for consistency between comprehensive domain ontologies. Often, new electric vehicles have a better power performance and, therefore, save time and money. For this reason, the structural, the conceptual and the rule layer have to use standardized data formats or should be involved in standardizations.

These use cases show that a knowledge base, which provides more functionality as a simple data base is useful in the context of logistics software and electric vehicles. In the next section, we present a cut-out of a possible knowledge base for European driving licence classes.

III. A Knowledge Base for European Driving Licence Classes

In this section, we exemplary introduce (parts of) a knowledge base for driving licence classes in the European Union. The use-case for this knowledge base is to derive drivers who can drive a specific vehicle type or vehicle types who can be driven by a specific driver.

In the European Community, a lot of driving licence classes exist. Figure 4 shows these classes and also a cut-out of their interdependencies. As we see, there are multiple

 ISBN: 978-1-61208-438-1

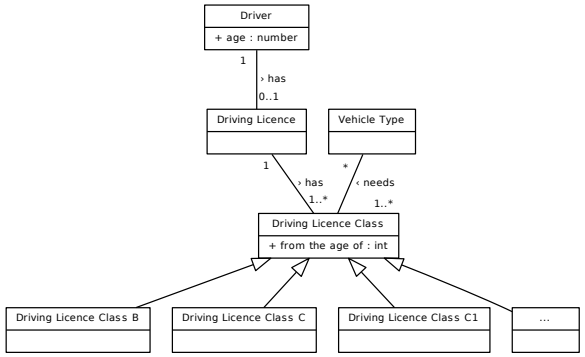

Figure 3. Entities in the knowledge base

interdependencies between those driving licence classes, which are difficult to know and to learn for an inexperienced user. For this reason, a knowledge base would be profitable for supporting logistic scheduler.

As argued in Section I, at first, we introduce the structural parts of our striven knowledge base. This structural layer consists of data descriptions, i.e., the description of the concepts. In our implementation, we have used an own data description language, which structures these concepts in groups, cardinalities, options, and entities as introduced in Döbrich and Heidel [5]. Those data descriptions contain also synonyms and textual descriptions as well as units. For a better readability, we use an UML 2.0 class diagram [6] at this point of view. That class diagram is illustrated in Figure 3.

The major concepts are the ones to represent a driver, a driving licence, a vehicle type, and driving licence classes. Naturally, several other concepts can be introduced to describe those concepts in more detail.

Furthermore, there are some connected attributes for these concepts. As start point, the concept *driver* consists of an age. A driver has up to one *driving licence*, which has at least one *driving licence class*. Such a class has a class-specific driver's age for which that class can be received of a person. Furthermore, a driving licence class is needed to drive several *vehicle types*. At last, there are some subtypes of driving licence classes like *driving licence class B*.

After we have build the structural layer, we have to introduce the conceptual layer. As mentioned in Section I, the conceptual layer contains relations between these concepts. Some of these relations are already defined in the structural layer. At first, for each class in the class diagram, we include an unary relation in our knowledge base. For this, in the following, we use (to represent arbitrary instances) variables d for representing a driver, c, c_1, c_2 for driving licence classes, a, a' for ages, dl for a driving licence, and finally v as an arbitrary instance of a vehicle type:

$$\begin{aligned} Driver(d)\ (= D(d)) \\ Driving\_Licence(dl)\ (= DL(dl)) \\ Driving\_Licence\_Class(c)\ (= DLC(c)) \\ Driving\_Licence\_Class\_C(c)\ (= DLC\_C(c)) \\ Driving\_Licence\_Class\_C1(c)\ (= DLC\_C1(c)) \\ Driving\_Licence\_Class\_B(c)\ (= DLC\_B(c)) \\ Vehicle\_Type(v)\ (= VT(v)) \end{aligned}$$

Afterwards, we need the explicit binary relations in the following equation, which are extracted from the associations and the attributes of the class diagram:

$$\begin{aligned} From\_The\_Age\_Of(c, a)\ (= FTAO(c, a)) \\ Age(d, a) \\ Has\_Driving\_Licence(d, dl)\ (= HasDL(d, dl)) \\ Has\_Driving\_Licence\_Class(dl, c)\ (= HasDLC(dl, c)) \\ Needs(v, c) \end{aligned}$$

Eventually, we introduce the subclass-associations of the class diagram as rules into our conceptual layer:

$$\begin{aligned} DLC\_C(c) \rightarrow DLC(c) \\ DLC\_C1(c) \rightarrow DLC(c) \\ DLC\_B(c) \rightarrow DLC(c) \end{aligned}$$

Now, we have a stable structural and conceptual layer for our knowledge base (for this cut-out). Like Figure 4 shows, there are many other interdependencies between driving licence classes and the involved concepts. To represent those dependencies, we have to create a rule layer, which contains additional information.

As basic for our rule layer, we want to check whether an arbitrary number ($x \in \mathbb{R}$) is greater than or equal to another number ($y \in \mathbb{R}$):

$$GEq(x, y) = "x \geq y"$$

Whether two driving licence class instances belong to the same driving licence class can be checked by $Equal$:

$$\begin{aligned} DLC\_C(c_1) \wedge DLC\_C(c_2) \rightarrow Equal(c_1, c_2) \\ DLC\_C1(c_1) \wedge DLC\_C1(c_2) \rightarrow Equal(c_1, c_2) \\ DLC\_B(c_1) \wedge DLC\_B(c_2) \rightarrow Equal(c_1, c_2) \end{aligned}$$

As shown in Figure 4, a driving licence class C includes the driving licence class C1, i.e., one driver having a class C can also drive vehicles, which needs class C1.

$$DLC\_C(c_1) \wedge DLC\_C1(c_2) \rightarrow Includes(c_1, c_2)$$

Furthermore, the same figure shows, that both classes C and C1 requires class B to be received. Thus, each driver with class C, for example, can also drive vehicles with class B.

$$\begin{aligned} DLC\_C(c_1) \wedge DLC\_B(c_2) \rightarrow Requires(c_1, c_2) \\ DLC\_C1(c_1) \wedge DLC\_B(c_2) \rightarrow Requires(c_1, c_2) \end{aligned}$$

Now, we can infer all driving licence classes, which are available with a single one when we merge the $Equal$, $Include$, and $Requires$ rule to a single $Contains$ rule:

$$\begin{aligned} Equal(c_1, c_2) \rightarrow Contains(c_1, c_2) \\ Includes(c_1, c_2) \rightarrow Contains(c_1, c_2) \\ Requires(c_1, c_2) \rightarrow Contains(c_1, c_2) \end{aligned}$$

As a kind of validation, we check whether a specific age is enough to allow the receiving of a driving licence class. Therefore, we overload the rule $Requires$:

$$DLC(c) \wedge FTAO(c, a') \wedge GEq(a, a') \rightarrow Requires(c, a)$$

To build a predicate that allows us the inference of the drivers who can drive a specific vehicle type, we have to derive the driving licence classes of a driver.

$$HasDL(d, dl) \wedge HasDLC(dl, c) \rightarrow HasDLC(d, c)$$

 ISBN: 978-1-61208-438-1

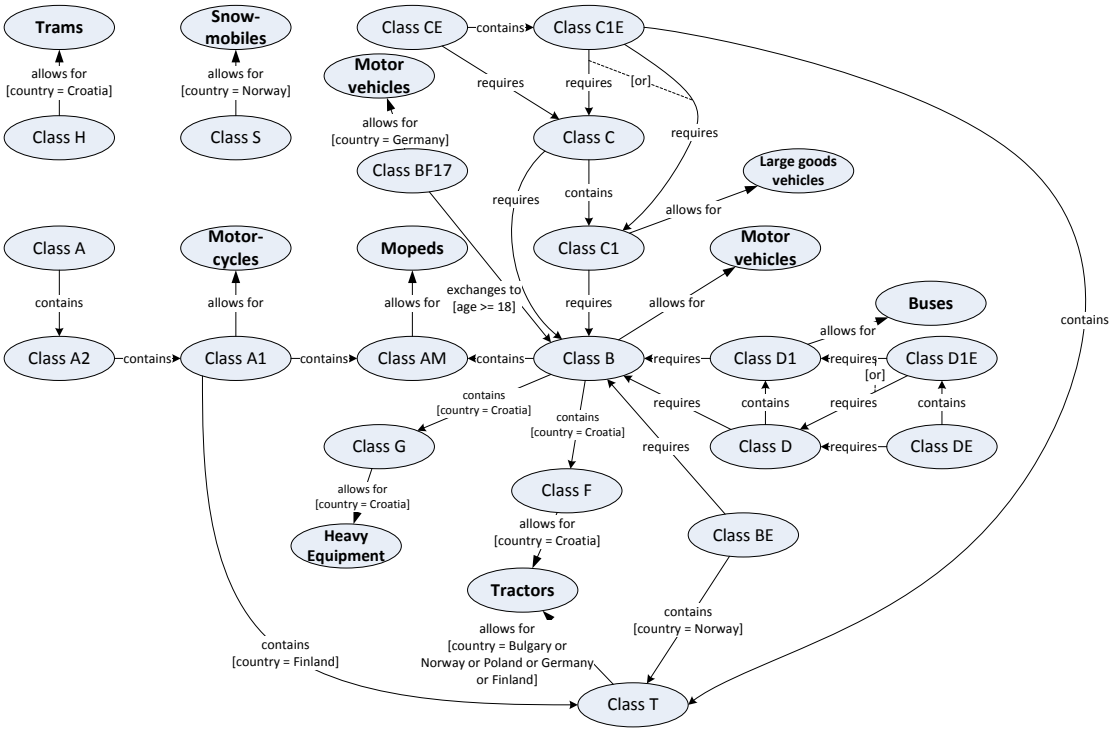

Figure 4. European driving licence classes and their interdependencies

To be sure that a driving licence class is valid for a driver, we introduce a rule that checks the age of the driver with regard to a specific driving licence class:

$$D(d) \wedge DLC(c) \wedge HasDLC(d,c) \wedge Age(d,a) \\ \wedge Requires(c,a) \rightarrow DHasDLC(d,c)$$

Finally, we can create a rule for our rule layer that allows us to infer all vehicle types that can be driven by a driver and all drivers that are allowed to drive a specific vehicle type:

$$VT(v) \wedge Needs(v,c_1) \wedge DHasDLC(d,c_2) \wedge Contains(c_2,c_1) \\ \rightarrow DCanDriveV(d,v)$$

With a simple implementation of our knowledge base, e.g., in Prolog, we can infer our wanted information by replacing a specific driver or vehicle with a variable, i.e., a place holder. That functionality can be adapted to logistics software, which automatically infer missing driving licence classes in documents and possible drivers for a tour. So, a knowledge base for electric vehicles is justifiable.

IV. Conclusion

In this work of progress paper, we have argued for a knowledge base for electric vehicles in inner-city logistics. Therefore, we have divided our knowledge base into three layers: the structural, conceptual, and rule layer. Furthermore, we have identified and explained some use cases for such a knowledge base and, finally, showed that it is possible to create such a knowledge base on an exemplary use case, which infers drivers, who are allowed to drive a specific vehicle type.

With the help of such a knowledge base and the presented use cases, it is possible to implement an intelligent software user interface for logistics that helps to create a consistent and complete data base. For example, if a company recruits a driver, the system can infer all of its implied driver licence classes although the driver stated the superordinate class $C1$.

In the future work, we have to extend our knowledge base with additional concepts, relations, and rules to allow more use cases and more safe inferences. Therefore, more knowledge has to be derived from the practice of using electric vehicles.

V. Acknowledgement

The project is funded by the German Federal Ministry for Economic Affairs and Energy, BMWi.

References

[1] V. Schau et al., “SmartCityLogistik (SCL) Erfurt: Deriving the main factors that influence vehicle range during short-distance freight transport when using fully electric vehicles,” in 10. GI/KuVS-Fachgespräch ”Ortsbezogene Anwendungen und Dienste”, pp. 101–108.

[2] S. Apel, T. M. Prinz, and V. Schau, “Challenging service extensions for electric vehicles in massively heterogenic system landscapes,” in Proceedings of the 7th Central European Workshop on Services and their Composition, ZEUS 2015, Jena, Germany, February 19-20, 2015., pp. 44–50.

[3] B. G. Buchanan et al., “Constructing an expert system,” Building expert systems, vol. 50, 1983, pp. 127–167.

[4] N. I. S. Organization, ANSI/NISO Z39.19 - Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies, N. I. S. Organization, Ed. National Information Standards Organization, May 2010, iSBN 978-1-880124-65-9.

[5] U. Döbrich and R. Heidel, “Datadriven Program Systems - a way out from interface chaos.” Informatik Spektrum, vol. 35, no. 3, 2012, pp. 190–203.

[6] J. Rumbaugh, I. Jacobson, and G. Booch, Unified Modeling Language Reference Manual, The (2nd Edition). Pearson Higher Education, 2004.

 ISBN: 978-1-61208-438-1