

# Ontology-based KP Development for Smart-M3 Applications

Aleksandr Lomov  
Petrozavodsk State University (PetrSU)  
Petrozavodsk, Russia  
lomov@cs.karelia.ru

## Abstract

Smart-M3 is an open-source platform that implements the smart space concept and maintains its content with ontological representation. SmartSlog is a tool for developing Smart-M3 knowledge processors (KPs) in terms of OWL classes, properties, and individuals. This paper describes new schemes of KP generating process and ways for improving KP development based on ontologies integration and general-properties. New generating schemes use Protege and ontologies meta-data. Protege is an ontology editor that is used to build the meta-data. The meta-data is used with SmartSlog generator as an extra information for KP generation. The ontology integration allows KP to work with knowledge represented by several ontologies in the smart spaces. The integration could be full (already supported by SmartSlog) or partial (added to SmartSlog based on new generation schemes) The partial integration allows to choose necessary entities from source ontologies. General-property allows to use a set of properties (slave-properties) as one property. Handlers are used with the general-property. Each handler define a logic of operations with slave-properties. The general-property approach now is under development.

**Index Terms:** Smart Spaces, OWL, Ontology, Smart-M3, SmartSlog.

## I. INTRODUCTION

Nowadays ontologies can be used not only for representing knowledge, but for constructing software [1]. The smart spaces approach states that the next generation of applications can be constructed on the basis of a common informational environment where heterogeneous participants dynamically share multi-source multi-domain data [2]. The Smart-M3 platform is an open-source implementation of the smart space concept. The data are represented with low-level RDF-triples and agents (Smart-M3 knowledge processors, KPs) cooperate sharing the common smart space. Developing KP using ontologies is a next step to improve constructing KP where the data and logic can be represented in one form.

SmartSlog ADK is a Smart Space ontology library generator [3] for Smart-M3 applications. The code is generated for a given OWL ontology; the application logic is programmed on top of the generated code in ontology terms. There are different operation that can be considered as an ontology reuse process [4], [5]. The KP generating process can be improved based on ontology manipulation.

SmartSlog generator now supports ontology integration: full and partial. To support partial integration the generator operates with ontological meta-data that is used to control generating process. To create the meta-data information for integration the Protege plug-in can be used. Protege is a free, open source ontology editor and knowledge-base framework.

The next solution to improve KP generation process is based on general-properties. Each general-property allows to map a set of properties to one general-property and to work with corresponding parts of the smart space. General-property contains a set of handlers.

Handler describes a logic of the operation under general-property. This solution is now under development.

The rest of the paper is organized as follows. Section II contains new schemes of the generation KP process. Section III describe integration of the ontologies and meta-data for partial integration. Section IV considers details of the general-property. Finally, Section V summarizes the paper.

## II. GENERATION PROCESS SCHEMES

SmartSlog contains CodeGen that implements generation of the ontology-dependent part of the KP library [3]. The CodeGen was able to operate only with ontologies. Now generation process scheme includes steps to operate with ontologies meta-data. The CodeGen extends base functionality of generation process by getting extra information from meta-data.

SmartSlog hasn't tool to edit ontologies and build meta-data. For these porpiose Protege [6] can be used. Protege can be extended by plug-ins to buld the meta-data of ontologies.

Fig. 1 shows two schemes of generating process with Protege and the plug-in. The first is already implemented and using for the KP development now. First scheme contains 3 main steps:

- 1) analyzing ontologies with Protege and plug-in;
- 2) creating meta-data;
- 3) generating ontology dependent part.

Creation of the meta-data is semiautomated process. The developer loads necessary ontologies with Protege and analyzes them with plug-in. After that the plug-in creates meta-data based on the input data. Then the developer uses the CodeGen. The CodeGen analyzes ontologies with the meta-data and creates an ontology dependent part of the KP library. Second scheme shows that the plug-in and the CodeGen can be combined in Protege and the developer works only with Protege.

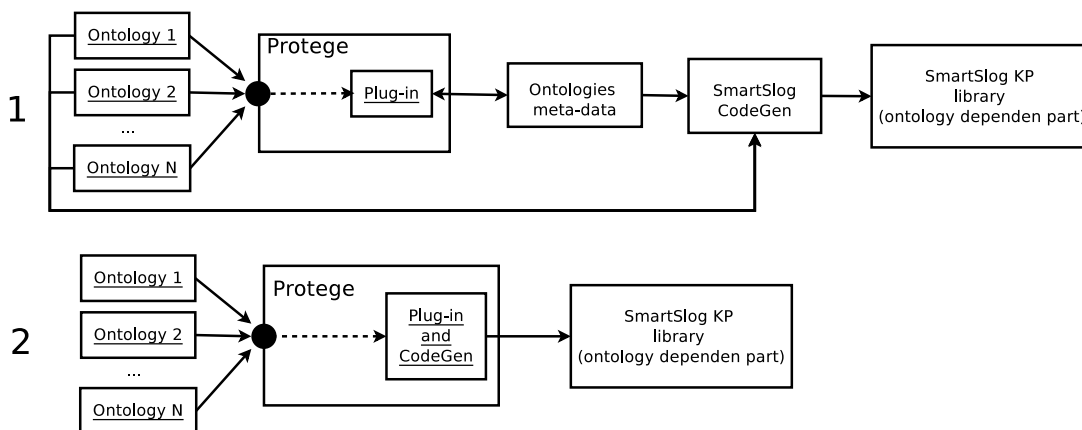


Fig. 1. Generation schemes

## III. ONTOLOGY INTEGRATION

The data in smart spaces can be described with several ontologies instead of a single big ontology. In this case a KP needs to know all ontologies that are used on the smart space side to operate with data. To achieve this an integration can be used.

Ontology integration is the process of generating a single ontology in one subject from two or more existing and different ontologies in different subjects [4], [7].

For KP development integration can be full or partial. With full integration several ontologies will be represented with one ontology. With partial integration entities from different ontologies can be selected and result ontology will contain only selected entities. The full integration was implemented in the first releases of the SmartSlog ADK. Now with new generation schemes and with using meta-data the partial integration is supported.

Fig. 2 shows ontology integration. First the CodeGen builds ontology-dependent part for KP based on three ontologies. With this ontology-dependent part a KP knows about all needed ontologies and can work with the data in smart spaces. For partial integration a filter-file is used. While analyzing ontologies the CodeGen uses meta-data from filter-file and includes needed entities to the result of generation.

The filter-file has a simple format and can be made by hand. It contains categories of entities with URIs:

```

--- Classes ---
http://www.xfront.com/owl/ontologies/camera/#Range
http://www.xfront.com/owl/ontologies/camera/#Viewer
--- Object properties ---
http://www.xfront.com/owl/ontologies/camera/#viewFinder
http://www.xfront.com/owl/ontologies/camera/#part
--- Data properties ---
http://www.xfront.com/owl/ontologies/camera/#f-stop
--- Object properties ---
http://www.xfront.com/owl/ontologies/camera/#compatibleWith
...

```

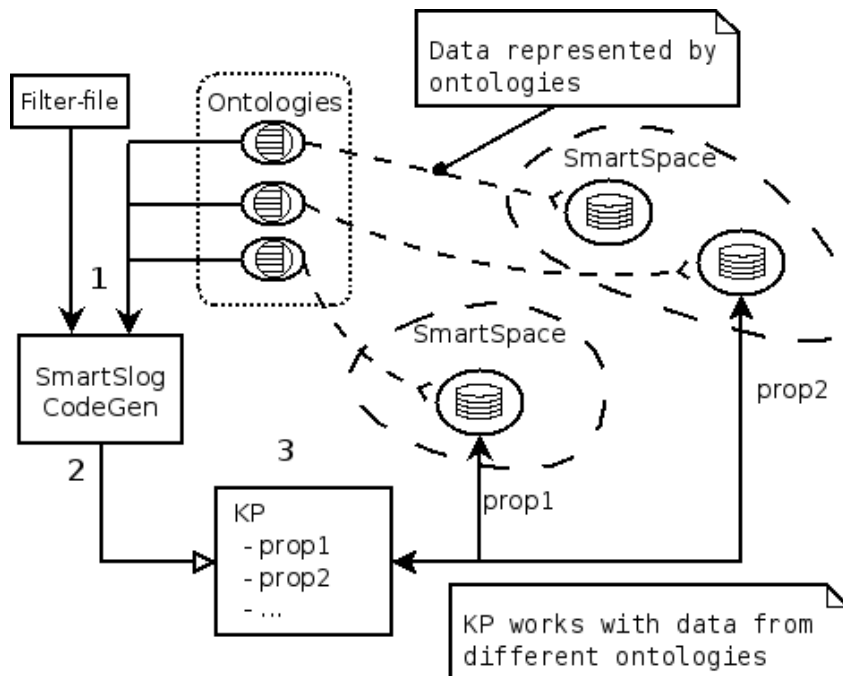


Fig. 2. KP based on ontology integration

To easily make the filter-file for several or large ontologies a Protege plug-in was implemented. Fig. 3 shows the Protege and the plug-in. The Protege provides an interface and

functionality to work with ontologies and the plug-in allows to choose needed entities: 1-classes, 2-object and data (not shown) properties.

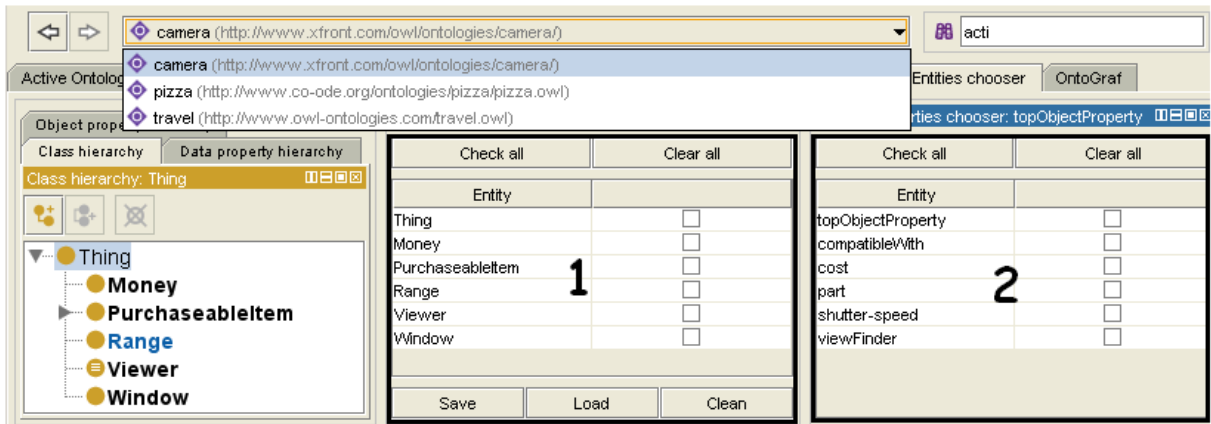


Fig. 3. Protege plug-in

#### IV. GENERAL-PROPERTY

It is possible to use some kind of the property to work with several data sources [8]. In SmartSlog this approach is a general-property (Fig. 4). The general-property is a virtual property and it exists only on the KP side. Operation with the general-property are reflected to real properties and values of properties are stored in or queried from all corresponding parts of the smart space. With SmartSlog this approach is extended with possibility define logic to perform operations under general-property.

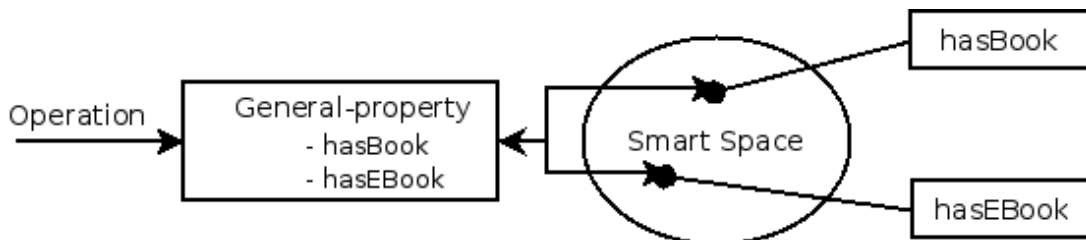


Fig. 4. General property

The general-property contains: 1) slave properties and 2) handlers. The slave properties is a set of properties from the real ontologies. Handlers are used to define logic of operation with the slave properties. The default handler translate operation with the general-property to the slave-properties. Other handler can be constructed by developer and improve or change base functionality of the general-property.

For example a KP needs to insert or remove persons and updates their number in the the smart space. The general-property can combine these properties and work with them using handlers. The developer sets needed properties and handlers to general-property and operates only with general-property.

Fig. 5 shows general-property that has 2 slave-properties: "hasPerson" and "count" and 2 handlers with logic for inserting and removing. There are 3 steps to perform an operation: 1) choose handler 2) prepare data to perform operation and 3) perform operations. One the first

step the general-property interrogates all handler of the possibility to perform the operation. On the second step, the handler prepares data: properties to insert, remove or update. On third step the general-property performs actions that were defined by handler also it tries to reduces number of requests to the smart space combining them.

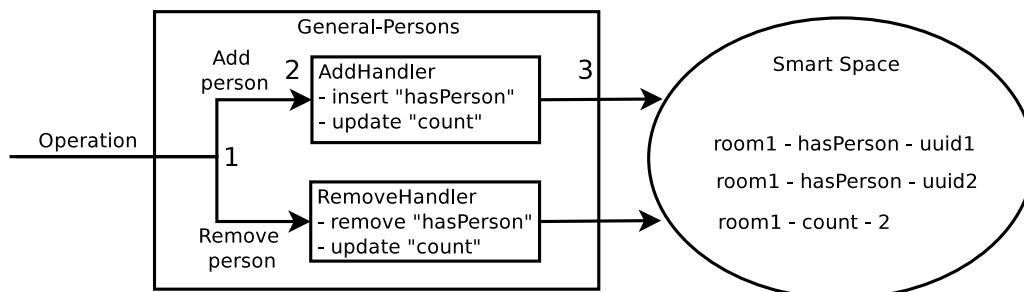


Fig. 5. General-property handlers

The general-property is under development now but prototype of API for C# SmartSlog version allows to use general-property as usual property. The code in the examples for familiarization only and in the release of SmartSlog some details or names could be changed. In the example 1 a general-property contains two real properties (line 2). On lines 4 and 6 two operation are perform under general-property (insert and remove). In this example a default handler is used and operations are translated to real properties. As a result for person two properties are inserted (line 4) to the smart space and in the end they are removed (line 6).

The example 2 shows how to create a handler (lines 2-10). To use own handler it needs to create a new classes based on 'Handler' class and set handler for general-property. 'MyHandler' class configures actions for general-property using methods of 'IHandlerConfig': ToInsert, ToUpdate, ToRemove. The 'IHandlerConfig' object contains slave-properties and information about operation under general-property this information can be used by the handler. On line 14 a new general-property with handler is created. The handler will configure operation and general-property perform needed actions. In this case 'MyHandler' for inserting operation sets 'hasFriend' property to insert, 'friendCount' property to update and for removing 'hasFriend' to remove, 'friendCount' to update.

---

**Algorithm 1** Examble of using general-property

---

```

Individual person = node.CreateIndividual(Man);
// Create a genelr property with default handler
GeneralProperty generalBook = new GeneralProperty(
    new List<Property>() hasBookProperty, hasEBookProperty);
// Insert general property: insert 'hasBookProperty' and 'hasEBookProperty'.
node.Insert(person, generalBook, "TestBook");
// Insert general property: remove 'hasBookProperty' and 'hasEBookProperty'.
node.Remove(person, generalBook, "TestBook");

```

---

## V. CONCLUSION

This paper studied the KP development based on ontological representation. SmartSlog allows to map an OWL ontology description to code (ontology library), abstracting the ontology and smart space access in KP application logic.

---

**Algorithm 2** Example of using general-property and handler

---

```
// Create a new Handler class for general-property
class MyHandler: Handler {
    protected override void ConfigureInsert(IHandlerConfig
config) {
        config.ToInsert(hasFriend, ..);
        config.ToUpdate(friendCount, ...);
    }
    protected override void ConfigureRemove(IHandlerConfig
config) {
        config.ToRemove(hasFriend, ..);
        config.ToUpdate(friendCount, ...);
    }
}
//...
// Create handler and set it for the property.
MyHandler myHandler = new MyHandler();
GeneralProperty friendGp = new GeneralProperty(
    new List<Property>{friendCount, hasFriend}, myHandler);
// Insert general property: insert 'hasFriend' and update 'friendCount'.
node.Insert(person, friendGp, friend1);
// Insert general property: remove 'hasFriend' and update 'friendCount'.
node.Remove(person, friendGp, friend2);
```

---

The SmartSlog generator simplifies KP development. With an integrated of ontologies a KP can work with several ontologies in the smart space. The integration can be full or partial. The generator supports partial integration using meta information about ontology. The meta-data can be build by Protege plug-in. The general-properties is used to combine properties form ontologies to operate with them using one property. Handlers of the general-property is used to define a logic of general-property. The general-properties support is under development in the SmartSlog now.

#### ACKNOWLEDGMENT

This research is a part of grant KA179 “Complex development of regional cooperation in the field of open ICT innovations” of Karelia ENPI programme, which is co-funded by the European Union, the Russian Federation and the Republic of Finland. The article was published with financial support from the Strategic Development Program of Petrozavodsk State University. We are grateful to the Open Innovations Association FRUCT for its support and R&D infrastructure. I would also like to thank Iurii A. Bogoiavlenskii and Dmitry G. Korzun for their feedback and expertise.

#### REFERENCES

- [1] J. Z. Pan, S. Staab, U. Aßmann, J. Ebert, and Y. Zhao, Eds., *Ontology-Driven Software Development*. Berlin: Springer, 2013.
- [2] I. Oliver, J. Honkola, and J. Ziegler, “Dynamic, localised space based semantic webs,” in *Proc. IADIS Int’l Conf. WWW/Internet 2008*. IADIS Press, Oct. 2008, pp. 426–431.
- [3] D. Korzun, A. Lomov, P. Vanag, J. Honkola, and S. Balandin, “Generating modest high-level ontology libraries for Smart-M3,” in *Proc. 4th Int’l Conf. Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2010)*, Oct. 2010, pp. 103–109.

- [4] N. Choi, I.-Y. Song, and H. Han, "A survey on ontology mapping," *SIGMOD Record*, vol. 35, pp. 34–41, Sep. 2006.
- [5] P. Bouquet, M. Ehrig, J. Euzenat, E. Franconi, P. Hitzler, M. Krötzsch, L. Serafini, G. Stamou, Y. Sure, and S. Tessaris, "D2.2.1 specification of a common framework for characterizing alignment," Knowledge Web Consortium, Tech. Rep., Feb. 2005. [Online]. Available: <http://knowledgeweb.semanticweb.org/semanticportal/deliverables/D2.2.1v2.pdf>
- [6] "Protege is a free, open source ontology editor and knowledge-base framework." [Online]. Available: <http://protege.stanford.edu/>
- [7] M. M. Taye, "Ontology alignment mechanisms for improving web-based searching," Ph.D. dissertation, Faculty of Technology De Montfort University, United Kingdom, England, 2009.
- [8] A. Maedche, B. Motik, L. Stojanovic, R. Studer, and R. Volz, "Ontologies for enterprise knowledge management," *IEEE Intelligent Systems*, vol. 18, pp. 26–33, Mar. 2003.