Ontology-based KP Development for Smart-M3 Applications

Aleksandr A. Lomov

Petrozavodsk State University
Department of Computer Science

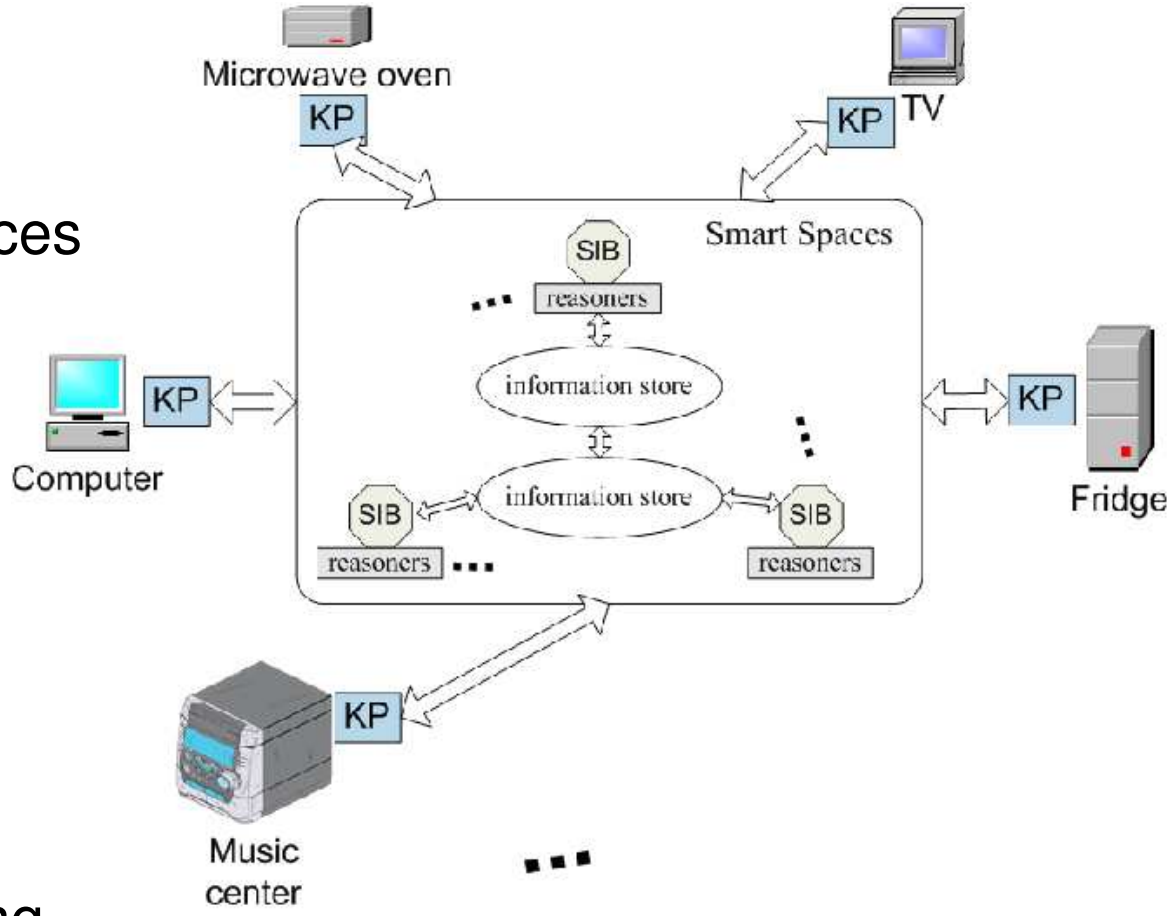
This project is supported by grant KA179 of Karelia ENPI - joint program of the European Union, Russian Federation and the Republic of Finland

13th FRUCT conference

April 22–26, Petrozavodsk, Russia

Smart-M3 Platform

- Smart spaces provide a shared view of resources
- Semantic information brokers (SIBs) maintain smart space content in low-level RDF triples
- Application consists of several knowledge processors (KPs) running on various devices
- Smart-M3: **M**ultidomain, **M**ultidevice, **M**ultivendor



Knowledge processors

Each KP is an agent sharing ad-hoc knowledge across numerous domains

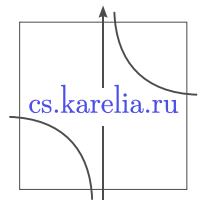
KP development approaches:

Low-level (RDF triple)

- KP $\leftrightarrow$ **RDF-Triples** $\leftrightarrow$ Smart Space

High-level (OWL ontology object)

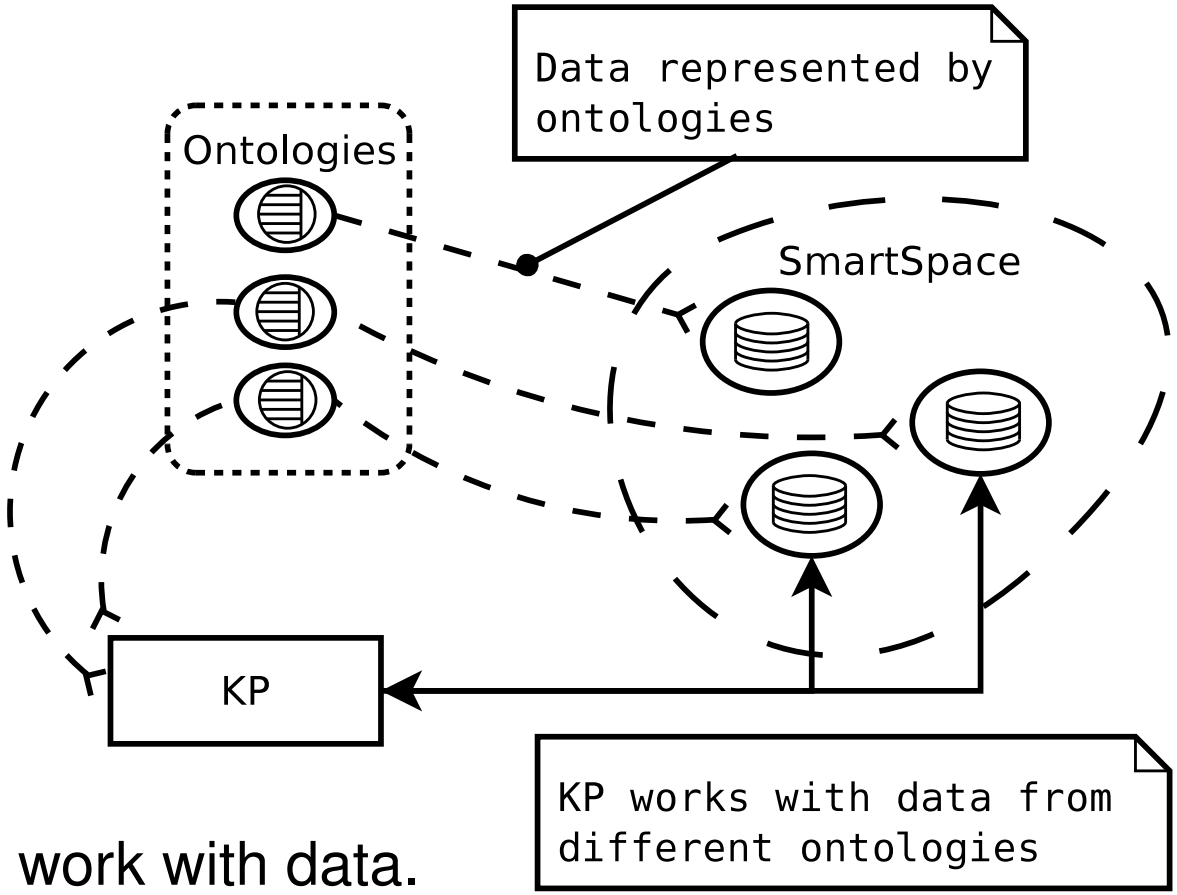
- KP $\leftrightarrow$ **Ontological objects** $\leftrightarrow$ RDF-Triples $\leftrightarrow$ Smart Space



SmartSpace access

Smart Room ontologies:

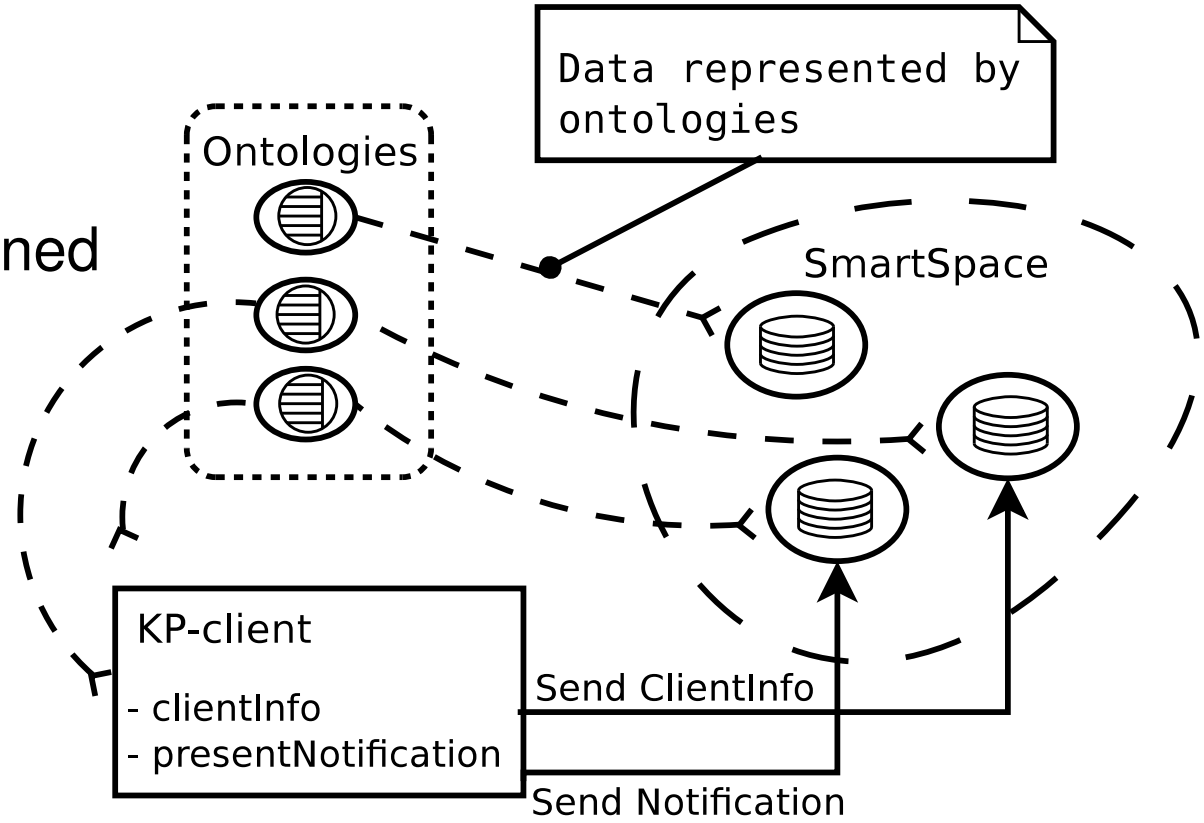
- Services-ontology
- Notification-ontology
- FOAF-ontology
- ...



KPs needs to know how to work with data.

Property dependencies

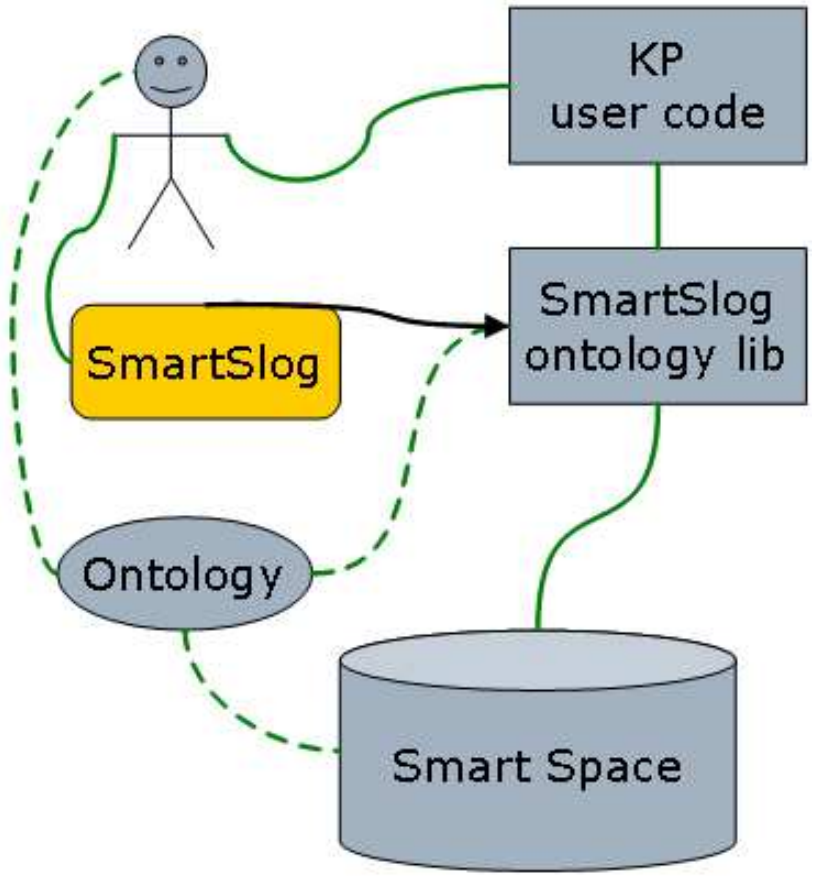
- Properties are combined in one logic block
- One operation to smart space



KPs works with several properties as with one.

SmartSlog SDK

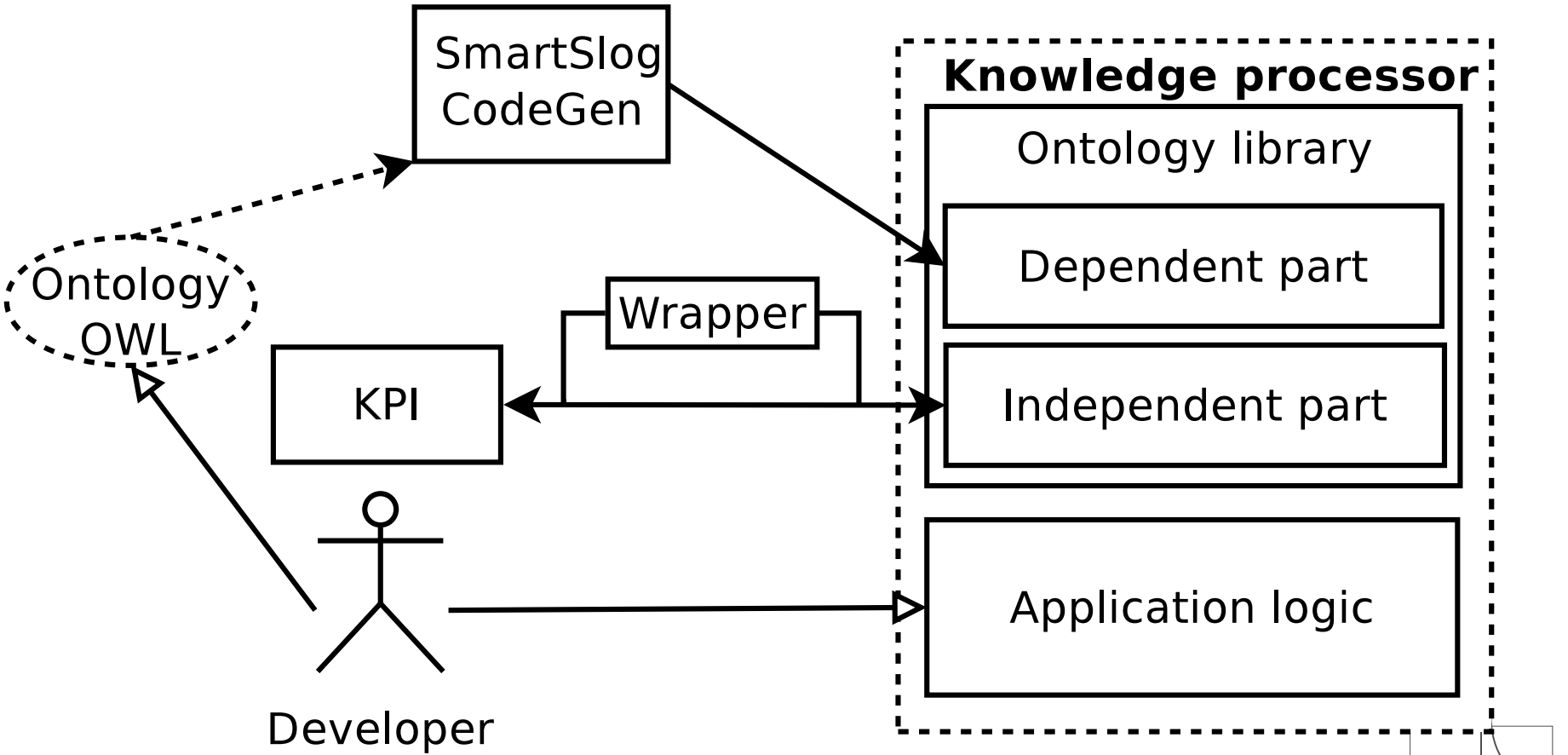
- Library generator for Smart Space ontology
- Mapping OWL to code (C, C#):
  - KP uses ontology library
  - ontology abstractions in API
  - low-level KPI is hided
- High-level communication primitives
  - session
  - knowledge patterns
  - subscription



KP programmer can think in abstract ontology terms!

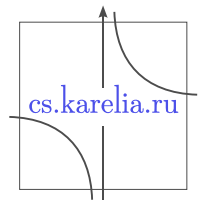
cs.karelia.ru

Generation scheme



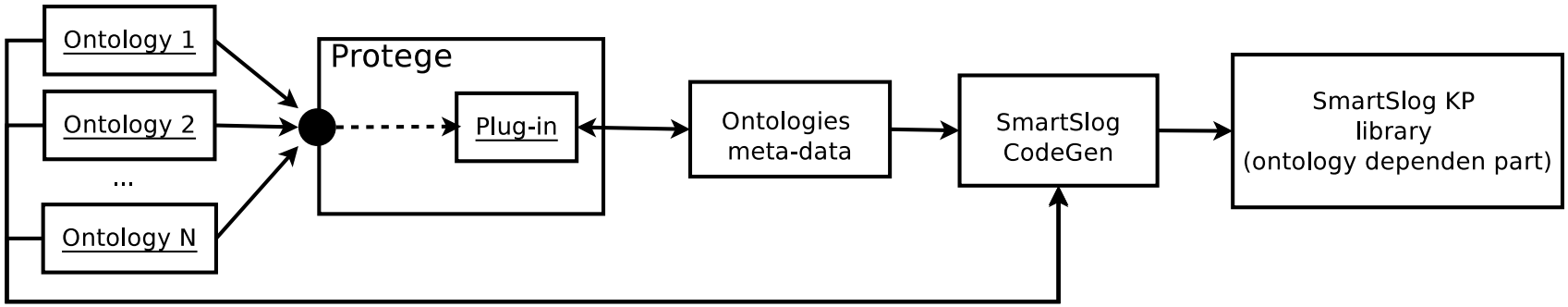
Code Generation

- Java-based CodeGen
- Mapping OWL Ontology to target code for KP
- Static templates/handlers scheme
- **Templates** are “pre-code” of data structures
  - implementation of ontology classes
  - implementation of properties for classes
  - tags $\langle \text{name} \rangle$ instead of proper ontology names
- **Handlers** transform templates into final code
  - Replacing tags with the names taken from the ontology
  - Executed when the ontology graph is analyzed

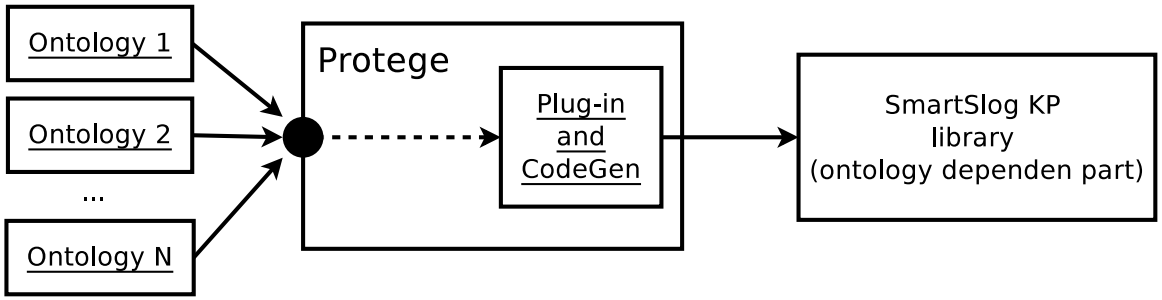


Extended schemes for KP generation

1



2

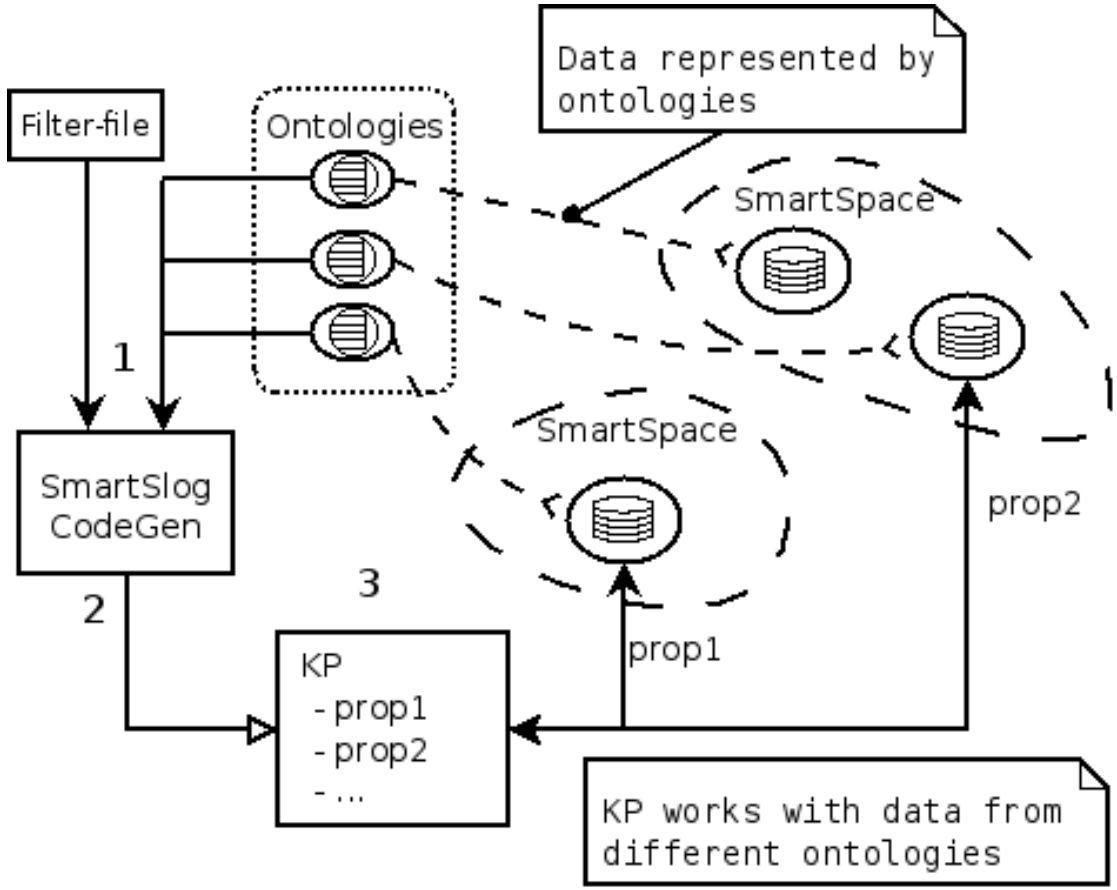


Protege is a free, open source ontology editor and knowledge-base framework

cs.karelia.ru

Ontology integration:

- Complete integration
- Partial integration



KP manipulates with several knowledge sets.

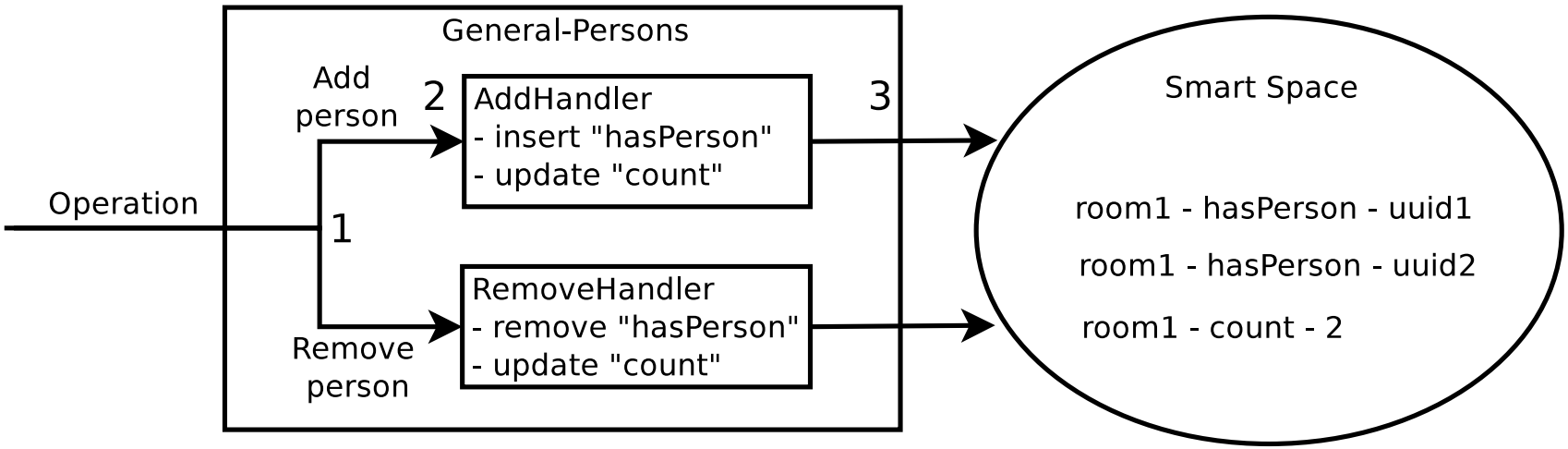
Partial integration is based on extended genereation scheme. A meta-data — filter-file (created with Protege plug-in).

cs.karelia.ru

General-property model

- Several properties are represented with one property
- Handlers defines operation logic
- Combining requests to the smart space

General-property is under construction now.



cs.karelia.ru

Conclusion

- SmartSlog SDK provides support for ontology-based KP development
  - Full ontology integration
  - Partial ontology integration (extended generation scheme, Protege plug-in)
  - General-property to work with several properties (prototype now)

Thank you!