Reasoning with LarKC

Michael Witbrock
Cycorp Europe
(and the LarKC consortium)
Semantics
Scientifically Relevant
Goals of LarKC

LarKC = Large Knowledge Collider

- **Build an integrated pluggable platform for web scale, incomplete, heterogeneous reasoning**
  - Support for parallelization, distribution, remote execution, data storage
  - Use existing plug-ins, develop new
  - Easy integration of components
  - Enables low cost experimentation

“Significant progress is sometimes made not by making something possible that was impossible before, but by substantially lowering the costs of something that was only possible before at high cost”
Overall approach of LarKC

• Powerful platform with convenient service access
  – communication, synchronization, registration
  – LarKC = “SPARQL endpoint on steroids”

• LarKC gives you:
  – very scalable data layer
  – standardised interfaces for combining components
  – utilities & infrastructure to abstract from remote deployment
  – Inference about plugins

• Most work happens in the plugins

• Three types of LarKC users:
  – people building plugins
  – people configuring workflows
  – people using workflows
What do we mean by:

- **reusable** components
- **reconfigurable** workflows
- provide **infrastructure** needed by all users:
  - storage & retrieval
  - registration of plugins
  - communication (plugin2datalayer, plugin2plugins)
  - synchronisation (anytime behaviour)
  - remote execution (abstracts from local/remote storage)
  - remote data-access (abstracts from local/remote invocation)
  - (will) provide instrumentation & measuring
  - (will) provide caching & data-locality

- integration of very **heterogeneous** components
  - heterogeneous data: unstructured text, (semi)structured data
  - heterogeneous code: Java, scripts, remote services ("wrap & integrate")

**LarKC = a platform for large scale reasoning**
Goals of LarKC, and where we are

- Scalable: > $10^9$ triples, lazy pipes
- Reconfigurable: plugins with standard API’s
- Open: Apache license
- heterogenous: TRANSFORM, wrappers
- experimentation: wrap & integrate
- allow incompleteness: IDENTIFY, SELECT
- enable distribution: plugin containers
- anytime behaviour: streaming APIs
- web-enabled: remote plugins & data
What Has Changed Since LarKC Started

• Scale is happening, and is important
  – academic work by Hitzler et al, Hendler et al, UT Dallas etc
  – commercial: OpenLink, Oracle, OntoText, BigData, …

• LOD accelerated growth and use (especially notable; Linked Govt Data)
The LarKC Platform Architecture 2.0

1) start

2) define Workflow

3) pass workflow parameters

4) workflow parameters

5) instantiate, pass workflow parameters, notify user after successful setup

6) Query

Workflow parameters (described in RDF)

Management Interface

LarKC entry point
* instantiates plug-in registry
* instantiates management interface (RDF endpoint)
* provides an executor per workflow (a single workflow may have multiple endpoints)

Executor

Endpoint (e.g. SPARQL)

User

Results

Data Layer

Transformer → Identifier → Selector → Decider → Reasoner → Decider

Backward chaining of plugin managers
What does a workflow look like?

Decider

Query Transformer -> Identifier -> Info Set Transformer -> Selecter -> Reasoner

Plug-in API

Workflow Support System

Plug-in Registry

Plug-in Manager

RDF Store

Query Transformer

Identifier

Info Set Transformer

Selector

Reasoner
What does a workflow look like?
WP5: The LarKC Platform

open & flexible

modular & extendable

distributed & scalable

LarKC@SourceForge
LarKC Codebase
Parallelisation
Plug-In API, Annotations and Plug-in Registry
Remote Execution
Documentation
Data Layer

Realised LarKC Platform Architecture 3.0
WP5: The LarKC Platform

- open & flexible
- modular & extendable

LarKC@SourceForge
LarKC Codebase
Plug-In API, Annotations and Plug-in Registry
Documentation
Realised LarKC Platform

Plugin Market
38 Plug-ins
Outline

- Final LarKC platform’s architecture
- Final Software Release
- User and Developer Support
- Summary of research and development activities
Outline

- Final LarKC platform’s architecture
- Final Software Release
- User and Developer Support
- Summary of research and development activities

13/12/2011
LarKC Platform’s Architecture

Overview

1. LarKC Platform's Architecture Overview

- Decider
- Identifier
- Reasoner
- Plug-in layer

- Developer extensions
- LarKC platform
- LarKC Data Layer

- LarKC middleware adapters/extensions

- Platform layer

- User environment
  - User desktop machine

- High-performance and Grid (Cloud) environment
  - High-performance and cluster systems
  - Native middleware
  - Public Desktop Grid
  - Native middleware
  - Volunteer resources
  - Native middleware

- Data Storage
  - RDF Doc
  - RDF Store

- Resource layer

13/12/2011
LarKC Platform’s Architecture

Overview

1. Market place
2. Development Wizards
3. Plug-in and application development support
4. Workflow Designer
5. Development Repository
6. Distribution Framework
7. Easy and transparent access to the storage, computation, and monitoring infrastructure
8. Parallelisation Support
9. Performance Analysis
10. End-user
11. Workflow designer
12. Plug-in developer
13. Applications
14. Workflows
15. Plug-ins
16. Deciders
17. Identifiers
18. Selectors
19. Transformers
20. Reasoners
21. LarKC platform
22. Management Interface
23. Plug-in Registry
24. Run-Time Environment
25. LarKC Data Layer
26. Plug-in Managers
27. Remote Managers
28. Monitoring Services
29. Web Server
30. High-performance computer
31. Computing Resources
32. Monitoring Infrastructure
33. LOD
34. LLD
35. Triple Stores
36. User domain
37. Platform domain
38. Infrastructure domain

13/12/2011
LarKC Platform’s Architecture

Overview

User domain

Market place
Development Wizards
Plug-in and application development support
Workflow Designer
Development Repository
Distribution Framework
Easy and transparent access to the storage, computation, and monitoring infrastructure
Parallelisation Support
Performance Analysis

Plug-in and application development support
Workflow designer
Plug-in developer

End-user

LarKC platform
Management Interface
Plug-in Registry
Run-Time Environment
LarKC Data Layer
Plug-in Managers
Remote Managers
Monitoring Services

Web Server
High-performance computing
Computing Resources
Monitoring Infrastructure
Triple Stores

Reasoners
Transformers
Selectors
Identifiers
Deciders
Plug-ins

Platform domain

Infrastructure domain

1 13/12/2011
Request from the previous release:

**simplify plug-in interfaces and development**

- New plug-in API – both input and output are represented in RDF
- Automatic parallelization (thread-based) on the statement level
- Data caching, instrumentation and event processing
- New Maven-based build and dependency management system
  - Improved version controlling and dependency management with Maven
  - Simplified procedure of new plug-in creation by introducing maven archetypes
Request from the previous release:

enhance the workflow functionality by enabling more complex configurations
Request from the previous release:
**enhance the workflow functionality by enabling more complex configurations**

1. LarKC Platform’s Architecture
2. User Domain: Workflows

Diagram:
- **Endpoint1**
- **Path1**
  - **GoogleIdentifier** (input)
  - **ResultParser**
  - **ResultTransformer**
  - **IrisReasoner**
  - **ResultFilter** (output)
- **Path2**
  - **GoogleImagesIdentifier** (input)
  - **ResultParser**
  - **ResultTransformer**
  - **IrisReasoner**
  - **ImageResultFilter** (output)
- **Endpoint2**
Using Plug-in Registry to Create Workflow

Represent Properties
- Functional
- Non-functional (e.g. QoS)
- WSMO-Lite Syntax

Logical Representation
- Describes role
- Describes Inputs/Outputs
- Automatically extracted using API
- Decider can use for dynamic configuration
  - Rule-based
  - Fast
Request from the previous release:
enhance workflow annotation, simplify data dependency specification
Request from the previous release:

enhance workflow annotation, simplify data dependency specification

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix larkc: <http://larkc.eu/schema#> .

# Define two plug-ins
_:plugin1 a <urn:eu.larkc.plugin.identify.TestIdentifier> .
_:plugin2 a <urn:eu.larkc.plugin.transform.TestTransformer> .

# Connect the plug-ins
_:plugin1 larkc:connectsTo _:plugin2 .

# Define a path to set the input and output of the workflow
_:path a larkc:Path .
_:path larkc:hasInput _:plugin1 .
_:path larkc:hasOutput _:plugin2 .

# Connect an endpoint to the path
_:ep a <urn:eu.larkc.endpoint.sparql> .
_:ep larkc:links _:path .
```
4. Final Workflow definition schema

- A novelty of the LarKC 3.0 platform is that one submit workflows in RDF to the LarKC platform.
- Workflows provided as RDF files to the executor.
- Introducing new elements:
  - Paths: a path represents an arbitrary linkage of plug-ins between the path input plug-in and its output plug-in.
  - EndPoint: Results are consumed via endpoints.
  - Endpoints are connected to a path via 'larkc:links.'
• **Improved LarKC Plug-in Annotation Language und Methodology**
  – Include non-functional QoS aspects of plug-ins.
  – Take into account Semantic Web services, Web API and RESTful services annotation models
  – Improved version of workflow annotation ontology.
Infrastructure

Execution Environment

Deployment Factor
Performance

```
<table>
<thead>
<tr>
<th>wsl#NonFunctionalParameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdf#subClassOf</td>
</tr>
<tr>
<td>larkc#Performance</td>
</tr>
<tr>
<td>larkc#hasPerformance</td>
</tr>
<tr>
<td>larkc#Plugin</td>
</tr>
<tr>
<td>larkc#hasEstimationInvocationTime</td>
</tr>
<tr>
<td>larkc#TemporalDuration</td>
</tr>
<tr>
<td>larkc#forExecutionEnvironment</td>
</tr>
<tr>
<td>larkc#ExecutionEnvironment</td>
</tr>
</tbody>
</table>
```
Scalability

[Diagram showing relationships between classes and properties related to scalability]
F-Measure, ROC Curve, AUC Curve
Temporal

```
larkc#TemporalEntity
  rdfs#subClassOf
  larkc#TemporalDuration

larkc#hasDurationValue                   larkc#hasUnitOfDuration
  xsd:int                                larkc#UnitOfDuration
                                           rdf#type
  larkc#Milliseconds
```
Cardinality

Limits
• LOS emerged from research toward the fusion of Linked Open Data and Semantic Web services:
  – More intuitive approach to Semantic Web services for Linked Data workers.
  – Relies on RDF, HTTP and SPARQL only.

• Principles:
  – Describe services' input and output as SPARQL graph patterns.
  – Communicate RDF by RESTful content negotiation.
  – The output should make explicit its relation with the input.
  – Make the lifting/mapping open as SPARQL CONSTRUCT queries.
Minimal Service model (SOA4All)
Plugin class is now sub-class of the service concepts as denoted by the Minimal Service Model.

- Plug-in inherits all the properties of a service offering much more annotation possibilities.
- This allows to semantically describe the structuring information which was initially given through the WSDL-files.
- The entire plug-in description in RDF.
3. Final Plug-in definition ontology

- Plug-ins now consume and produce RDF only, and no information sets or arbitrary formats so the description of data types is now obsolete.
- In the plug-in annotation, parameters are only typed and linked to a graph pattern, as in Linked Open Services.

```
<urn:eu.larkc.identifier.HomeAddress> {
    <http://example.com/johnd#me> v:adr [ 
        a v:Home ;
        v:street-address "Musterstrasse 1" ;
        v:postal-code "6020" ;
        v:locality "Innsbruck" ;
        v:country-name "Austria"
    ] .
```
3. Final Plug-in definition ontology

```
<urn:eu.larkc.plugin.reasoner.IRISReasoner> rdfs:subClassOf larkc:Plugin ;
  sawsd1:modelReference ex:OWL2RLReasoner ;
  larkc:hasInputParameter <urn:eu.larkc.reasoner.FactBase> ;
  larkc:hasOutputParameter <urn:eu.larkc.reasoner.KnowledgeBase> .

<urn:eu.larkc.reasoner.FactBase>
  larkc:hasPattern "\[] rdf:value ?filename" .
<urn:eu.larkc.reasoner.KnowledgeBase>
  larkc:hasPattern "\[] rdf:value ?filename" .

# example classification
ex:Reasoner a wsl:FunctionalClassificationRoot .
ex:OWL2Reasoner rdfs:subClassOf ex:Reasoner .
ex:OWL2RLReasoner rdfs:subClassOf ex:OWL2Reasoner .
```
4. Final Workflow definition schema

```prolog
_:p1 a <urn:at.sti2.larkcplugins.selector.InnsbruckTouristBoardReader> .
_:p2 a <urn:at.sti2.larkcplugins.selector.YellowPagesAustriaReader> .
_:p3 a <urn:eu.larkc.plugin.reasoner.IRISReasoner> ;
  larkc:hasParameter [
    larkc:binds <urn:eu.larkc.reasoner.FactBase> ;
    rdf:value "file://localhost:80/facts/f1.rdf"
  ],
  [
    larkc:binds <urn:eu.larkc.reasoner.FactBase> ;
    rdf:value "file://localhost:80/results/kb.rdf"
  ]
  larkc:hasInputBehavior larkc:All ;
  larkc:connectsTo _:p4 .
_:p4 a <urn:at.sti2.larkcplugins.decider.RestaurantCoverageDecider> .
_:p1 larkc:connectsTo _:p3 .
_:p2 larkc:connectsTo _:p3 .

<urn:eu.larkc.endpoint.SPARQUEndpoint> rdfs:subClassOf larkc:Endpoint .
_:e a <urn:eu.larkc.endpoint.SPARQUEndpoint> ;
  larkc:links [
    larkc:hasInput _:p1 , _:p2
    larkc:has0uput _:p4 ] .

larkc:All a larkc:Behavior .
```
Request from the previous release:
provide user-friendly tools for Workflow Design
Request from the previous release: *provide more end-points for different use cases*

<table>
<thead>
<tr>
<th>End-point's name</th>
<th>Input</th>
<th>Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SparqlEndpoint</td>
<td>Query in SPARQL syntax</td>
<td>standard SPARQL result XML format</td>
<td>A SPARQL endpoint enables users (human or other) to query a knowledge base via the SPARQL language. Results are typically returned in one or more machine-processable formats. A SPARQL endpoint is meant to be used by the machines as an interface towards a knowledge base.</td>
</tr>
<tr>
<td>QueryEndpoint</td>
<td>Query in any syntax</td>
<td>Returns the result of the workflow in RDF/XML syntax</td>
<td>The ?QueryEndpoint takes the query in any syntax, which means that the endpoint does not perform any checks on the query. It is assumed that the workflow can handle the query.</td>
</tr>
<tr>
<td>PushEndpoint</td>
<td>HTTP request with a parameter which points to the downloadable RDF file</td>
<td>/</td>
<td>Downloads the given RDF file when requested and passes it on to the workflow</td>
</tr>
<tr>
<td>ActiveEndpoint</td>
<td>HTTP request without any parameters</td>
<td>HTML Web page</td>
<td>This endpoint is a part of Active workflow and is very specific. It is not meant to be used with any other workflow. Anyhow, it can be used as an example.</td>
</tr>
<tr>
<td>TestEndpoint</td>
<td>Query in any syntax</td>
<td>Returns the result of the workflow in RDF/XML syntax</td>
<td>This endpoint is used for testing purpose. If the endpoint archetype is used to create an endpoint, the generated endpoint implements a ?TestEndpoint to give the developer an impression on how to implement the functionality.</td>
</tr>
</tbody>
</table>
Request from the previous release: **provide more end-points for different use cases**

<table>
<thead>
<tr>
<th>End-point's name</th>
<th>Input</th>
<th>Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SparQL Endpoint</td>
<td>Query in SPARQL syntax</td>
<td>standard SPARQL result XML format</td>
<td>A SPARQL endpoint enables users (human or other) to query a knowledge base via the SPARQL language. Results are typically returned in one or more machine-processable formats. A SPARQL endpoint is meant to be used by the machines as an interface towards a knowledge base.</td>
</tr>
<tr>
<td>PushEndpoint</td>
<td>HTTP request with a parameter which points to the downloadable RDF file</td>
<td>/</td>
<td>Downloads the given RDF file when requested and passes it on to the workflow.</td>
</tr>
<tr>
<td>ActiveEndpoint</td>
<td>HTTP request without any parameters</td>
<td>HTML Web page</td>
<td>This endpoint is a part of Active workflow and is very specific. It is not meant to be used with any other workflow. Anyhow, it can be used as an example.</td>
</tr>
<tr>
<td>TestEndpoint</td>
<td>Query in any syntax</td>
<td>Returns the result of the workflow in RDF/XML syntax</td>
<td>This endpoint is used for testing purpose. If the endpoint archetype is used to create an endpoint, the generated endpoint implements a ?TestEndpoint to give the developer an impression on how to implement the functionality.</td>
</tr>
</tbody>
</table>

In the final platform’s architecture the end-points are pluggable, i.e. handled by the platform similarly as plug-ins.
1 LarKC Platform’s Architecture

Overview

LarKC Platform’s Architecture Overview

- Market place
- Development Wizards
- Plug-in and application development support
- Workflow Designer
- Development Repository
- Distribution Framework
- Easy and transparent access to the storage, computation, and monitoring infrastructure
- Parallelisation Support
- Performance Analysis

LarKC platform

Management Interface

- Plug-in Registry
- Run-Time Environment
- Monitoring Services
- LarKC Data Layer

Applications

- Workflows
- Plug-ins

Deciders
Identifiers
Selectors
Transformers
Reasoners

End-user
Workflow designer
Plug-in developer

User domain
Platform domain
Infrastructure domain

LarKC Final Review Meeting, Vienna

13/12/2011
LarKC Platform’s Architecture
Platform Domain: Main Subsystems

- **Mgmt Interface**
- **Plug-in Registry**
- **LarKC RTE**
- **Plug-in Manager**
- **Data Layer**
- **Distribution**

- Workflow submission
- Management of plug-ins
- Initialization and invocation of workflows
- Plug-in execution
- Data management
- Flexible remote plug-in
a front-end to LarKC
implemented as a RESTful service
allows the submission/monitoring/deletion of workflows
supports N3 and RDF/XML workflow representations
equipped with additional features for retrieving
– all registered plug-ins from the platform
– configuration templates of all supported remote hosts
– all registered end-points
accessible via a simple HTML interface
LarKC Platform’s Architecture

Platform Domain: Runtime Environment

- improved workflow’s life-cycle management (initialisation, execution, and termination)
- coupled with Plug-in Registry and Plug-in Managers
- dataflow control (splitting and merging data, conditional loops, workflow branching, etc.)
• The Data Layer serves for LarKC plug-ins:
  – storage, retrieval and light-weight inference on top of large volumes of data
    • Reference implementation of ORDI data model
    • Retrieval data exposed via standard SPARQL endpoints
    • Configurable forward-chaining reasoning OWL2-RL
  – automates the exchange of RDF data by reference and by value
  – offers other utility tools to manage data (e.g. data streaming, querying remote data)
Used Concepts in the Data Model

Labelled groups of statements

NG1
NG2
NG3
NG4
NG5
## Supported Sets of Statements

<table>
<thead>
<tr>
<th>RDF data types</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set of statement</td>
<td>RDF statements</td>
<td>s1, p1, o1, ng1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>s2, p2, o2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>s3, p3, o3, ng3, {group1}</td>
</tr>
<tr>
<td>RDF graph</td>
<td>Named graph</td>
<td>s1, p1, o1, ng1, {group1}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>s2, p2, o2, ng1, {group2}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>s3, p3, o3, ng1</td>
</tr>
<tr>
<td>Dataset</td>
<td>SPARQL dataset</td>
<td>s1, p1, o1, ng1</td>
</tr>
<tr>
<td></td>
<td>represents a collection of graphs</td>
<td>s2, p2, o2, ng2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>s3, p3, o3, ng3</td>
</tr>
<tr>
<td>Labelled group of statements</td>
<td>RDF group of statements</td>
<td>s1, p1, o1, ng1, {group1}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>s2, p2, o2, ng2, {group1}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>s3, p3, o3, ng3, {group1}</td>
</tr>
</tbody>
</table>
# Different Implementations

<table>
<thead>
<tr>
<th>Name</th>
<th>Purpose</th>
<th>Cost to pass $10^6$ RDF statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set of statements</td>
<td>Services to return RDF data (e.g., SPARQL construct and describe queries)</td>
<td>350MB memory</td>
</tr>
<tr>
<td>Named graph in memory</td>
<td>Pass named graphs between plug-ins</td>
<td>350MB memory</td>
</tr>
<tr>
<td>HTTP published RDF graph</td>
<td>Load chunks of RDF from URL</td>
<td>few KBs + time to load the data</td>
</tr>
<tr>
<td>Named graph retrieved from a SPARQL endpoint</td>
<td>Pass named graphs using a SPARQL endpoint</td>
<td>few KBs + time to transfer the data + to filter the dataset with SPARQL</td>
</tr>
<tr>
<td>SPARQL dataset</td>
<td>Constrain RDF data exposed by a SPARQL endpoint with a list of graph names</td>
<td>few KBs + time to transfer the data + to filter the dataset with SPARQL</td>
</tr>
<tr>
<td>Labelled group of statements</td>
<td>Pass arbitrary sets of RDF statements between plug-ins (e.g., graph priming)</td>
<td>few KBs + time to associate the statements + time to filter the labelled group</td>
</tr>
</tbody>
</table>
LarKC Platform’s Architecture

Overview

- Market place
- Development Wizards
- Plug-in and application development support
- Workflow Designer
- Development Repository
- Distribution Framework
- Easy and transparent access to the storage, computation, and monitoring infrastructure
- Parallelisation Support
- Performance Analysis

Applications

Workflows

Plug-ins

Deciders  Identifiers  Selectors  Transformers  Reasoners

LarKC platform

Management Interface

Plug-in Registry

Run-Time Environment

LarKC Data Layer

Plug-in Managers

Monitoring Services

Remote Managers

Web Server

High-performance computer

Monitoring Infrastructure

Computing Resources

Triple Stores

RDF Doc

RDF Doc

LOD

LLD

End-user

Workflow designer

Plug-in developer

User domain

Platform domain

Infrastructure domain
LarKC Platform’s Architecture

Resource Domain: Distributed Execution

Decider

Plug-in Manager
- Plug-in API
- Identifier

Plug-in Manager
- Plug-in API
- Remote Launch Mechanism

Plug-in Manager
- Plug-in API
- Selector

Plug-in Manager
- Plug-in API
- Identifier

queue control system

http://
powered by JEE technology

ssh://
powered by GAT

13/12/2011
LarKC Platform’s Architecture

Resource Domain: Distributed Execution

Workflow branching

Plug-in parallelisation

Multi-threading
MPI
MapReduce
LarKC Platform’s Architecture

Resource Domain: Distributed Execution

Local host

- Input queue
- SoS Serializer
- Plug-in Manager
- File
- Remote Launch Mechanism
- Remote Agent
- Remote Invocation
- Poststage

Remote host

- Prestage
- Remote Invocation
- Poststage

Network

- Remote Agent
- Plug-in API
- Identifier
- SoS Deserializer
- Output queue

13/12/2011
# Workflow Description
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix larkc: <http://larkc.eu/schema#> .

# Define plug-ins
_:plugin1 a <urn:eu.larkc.plugin.LLDReasoner> .
_:plugin1 larkc:runsOn _:host1 .

# Define hosts
_:host1 a <urn:eu.larkc.host.Tomcat> .
_:host1 larkc:hostType larkc:JEE .

# Define a path to set the input and output of the workflow
_:path a larkc:Path .
_:path larkc:hasInput _:plugin1 .
_:path larkc:hasOutput _:plugin1 .

# Connect an endpoint to the path
_:ep a <urn:eu.larkc.endpoint.sparql.SparqlEndpoint> .
_:ep larkc:links _:path .
# Workflow Description
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix larkc: <http://larkc.eu/schema#> .

# Define plug-ins
_:plugin1 a <urn:eu.larkc.plugin.LLDReasoner> .
_:plugin1 larkc:runsOn _:host1 .

# Define hosts
_:host1 a <urn:eu.larkc.host.Tomcat> .
_:host1 larkc:hostType larkc:JEE .

# Define a path to set the input and output of the workflow
_:path a larkc:Path .
_:path larkc:hasInput _:plugin1 .
_:path larkc:hasOutput _:plugin1 .

# Connect an endpoint to the path
_:ep a <urn:eu.larkc.endpoint.sparql.SparqlEndpoint> .
_:ep larkc:links _:path .
WP11: Instrumentation
<table>
<thead>
<tr>
<th>Outline</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Final LarKC platform’s architecture</td>
<td></td>
</tr>
<tr>
<td>• Final Software Release</td>
<td></td>
</tr>
<tr>
<td>• User and Developer Support</td>
<td></td>
</tr>
<tr>
<td>• Summary of research and development activities</td>
<td></td>
</tr>
</tbody>
</table>
Open Apache 2.0 license
Previous early adopters workshops @ ESWC ’09,10 and ISWC ’09
- participants modified plug-ins, modified workflows

Standard Open Environment:
Java, subversion, packaged release, command line build, or eclipse
Main Features (1)

New (plain) plug-in registry

• light-weight plug-in loading and thus very low platform's start-up time
• adding/removing/replacing the plug-ins during the platform run, so that no restart is needed
• the plug-in registry from the previous releases, which is based on the Cyc's Knowledge Base with extended reasoning capabilities, can still be used if the platform is started with the "-cyc" option

New end-point support system

• end-points became plug-able and can be added to the plug-in on-the-fly by means of the special registry, similar as plug-ins
• added maven archetype that enables easy creation of new end-points
Improved Management Interface

- retrieval of the list of registered workflows
- special field for specification of the end-point's URL for query submission

Workflow Designer GUI, simplifying the use of LarKC

- lowering entry barrier for adopters

Improved Distributed Execution Framework

- the remote host's properties can be specified in both ways - directly in the workflow description and in a separate file
- fully supported by the Workflow Designer

Data caching, instrumentation and event processing

Final Software Release

Main Features (2)

2 Final Software Release

Main Features (2)
Final Software Release Documentation

New End-User Guide

LarKC
The Large Knowledge Collider
a platform for large scale integrated reasoning and Web

FP7 – 215535

Enhanced User & Developer Manual

LarKC
The Large Knowledge Collider
a platform for large scale integrated reasoning and Web

FP7 – 215535

Quick-Start Conference Tutorials

D5.4.3
Final Release of the LarKC Platform

Coordinator: Alexey Cheptsoy (HRI)
With contributions from: Matthias Assel, Ax (HLRS), Christoph Fuchs, Norbert Lanzanasto (ST)
Vasil Mointchev (ONTO), Spyros Koutoulas (VL)
Bradesko, Blaz Fortuna (CVC), Ioan Tom, S (Softgreek)
Quality Assessor: Alexey Cheptsoy (HLRS)
Quality Controller: Alexey Cheptsoy (HRI)

Document Identifier: LarKC/2008/D5.4.3
Date: September 29, 2011
Version: 1.0
Release Deliverable: LarKC EU ST-2008-215535
Status: Final
Distribution: public
Final Software Release
Software Management System

- Software releases
  - Platform binaries
  - User and developer guide

- Source code
  - Maven repository
  - Subversion repository

- Quality management
  - Hudson
  - Nightly build reports
2 Final Software Release
Software Management System

- Software releases
  - Platform binaries
  - User and developer guide

- Source code
  - Maven repository
  - Subversion repository

- Quality management
  - Hudson
  - Nightly build

Looking for the latest version? Download LarKC_v3.0.zip (33.0 MB)
Outline

• Final LarKC platform’s architecture
• Final Software Release
• User and Developer Support
• Summary of research and development activities
Final Software Release

User and Developer Support

- Bug Tracker
  - Bug Tracker
  - Forums
    - Developers (plug-ins)
    - Users (workflows)
  - Mailing Lists
    - larkc-dev-support@lists.sourceforge.net
    - larkc-user-support@lists.sourceforge.net

13/12/2011
Outline

• Final LarKC platform’s architecture
• Final Software Release
• User and Developer Support
• Summary of research and development activities
Summary

Software Development: Releases

- **GForge Alpha 1.0** (2009)
  - First prototype, introduced Plug-in API, implemented core platform’s services

- **SourceForge v2.0** (2010)
  - Ported to Maven, updated architecture, separated workflow specification and execution

- **SourceForge v2.5** (2011)
  - More tight integration of platform services, user-level tools (Workflow Designer, Wizards)

- **SourceForge v3.0** (2012)
  - The final release, full implementation of the Platform Architecture

- **Marketplace** (2012+)
  - plug-in farm
Download statistics for the Platform

- 816 downloads (10 Months)
- 459 downloads (3 Months)

35 downloads for v.3.0 in (10 days!!!)
Research Topics
WP2: Retrieval and Selection

Achieved: experimental setup for applying IR, Cognitive Memory and Spreading Activation methods for the problem of selection/subsetting/RDF search on large scale

Demonstrated:

- Scalable, parallelised selection strategies:
  - Cognitive Memory:
    - Interests-based selection use cases for medical researchers and computer scientists
  - Sub-setting based on statistical semantics applied to text and then linked to RDF Graphs
  - Statistical semantics applied directly to large RDF graphs (RDF molecules)

- Cutting-edge work on parallelisation
  - Cognitive Memory Spreading activation demonstrated on CUDA “GPU” cards
Part of broader effort led by Ontotext, WICI (Beijing), Max Planck (Berlin) to apply techniques derived from human psychology to web-scale reasoning

- The performance of the spreading activation tasks varies considerably depending on the parameters of the process

- As a reference point use the following result: it takes 7 seconds to activate about 7 thousand nodes after spreading of activation from resource http://dbpedia.org/resource/Berlin with decay factor 0.25.

- Queries on the “primed” or “selected” part of a dataset run up to 20 times faster and return only focussed results

    | Query performance after selection through priming | Query 1 | Query 2 |
    |-----------------------------------------------|---------|---------|
    | # of active nodes after graph priming          | 368,381 | 368,413 |
    | # of results (time for eval.) before priming   | 2,174 (87 ms) | 163,438 (518 ms) |
    | # of results (time for eval.) after priming    | 23 (5 ms) | 530 (47 ms) |
    | Reduction ratio for the number of results      | 94.5    | 308.4   |
C-SPARQL extends SPARQL to support continuous queries,
queries registered over RDF data streams and then continuously executed.
Queries consider windows, i.e. the most recent triples of such streams, observed while data is continuously flowing.
Stream Reasoning is Ubiquitous

• “Is a traffic jam going to happen in this highway? And is it convenient to reallocate travelers based upon the forecast?”
• “By looking at the click stream coming from a given IP, can we notice the shifts of interest of the person behind the computer?”
• “Which contents of the news Web portal are attracting more attention? Which navigation pattern would lead readers to other news related to those contents?”
• “Are trends in medical records indicative of any new disease spreading in given parts of the world?”
• “Where are all my friends meeting?”
• “In the financial context, can we detect any intraday correlation clusters among stock exchange?”
WP3: Abstraction and Learning

Machine learning for the Semantic Web

Traffic-Aware Routing Showcase

Social Media Analytics using C-SPARQL and SUNS

Legend
- C-SPARQL query
- SPARQL with Probability
- RDF graph
- RDF stream
- data stream
WP4: MaRVIN Architecture

• MaRVIN is:
  – a platform for distributed RDF(S) reasoning
  – a platform for processing lots of RDF data

• MaRVIN scales by:
  – distributing computation over many nodes
  – approximate (sound but incomplete) reasoning
  – anytime convergence (more complete over time)
WP 4: Reasoning and Deciding - WebPIE

- Uses Cluster/MapReduce to distribute computation
- RDFS and the OWL-horst rule-sets
- Full forward inference over large datasets

![Graph showing execution time versus number of nodes with data points for 8, 16, 32, and 64 nodes and corresponding runtime speedup values.]
WP 4: Reasoning and Deciding - WebPIE

- Uses Cluster/MapReduce to distribute computation
- RDFS and the OWL-horst rule-sets
- Full forward inference over large datasets

- First in SemWeb community (to break the 100B triple barrier)
- Now on Amazon EC2 for push-button use by 3rd parties.

Reasoning over Life Data
Use transitivity to find all types of cancer (long transitive chains)
WP5: Parallelisation Patterns

Plug-in parallelisation

- Multi-threading
- MPI
- MapReduce

Workflow branching
WP5: Parallelisation Patterns

- Transformer Plug-in (WP6)
  - Speed-up: ≈ 4

- MPI parallelisation of Random Indexing code (WP2,7)
  - Speed-up: ≈ 20-40

Plug-in parallelisation

- Multi-threading
- MPI
- MapReduce
WP5: Platform Distributed Execution Support

Decider

Plug-in Manager
- Plug-in API
- Identifier

Plug-in Manager
- Plug-in API
- Remote Launch Mechanism

Plug-in Manager
- Plug-in API
- Selector

Plug-in Manager
- Plug-in API
- Identifier

http:// powered by JEE technology

ssh:// powered by GAT
WP6 - Urban Computing Use Case

Road Sign Management Demonstrator
Demosntrates Reasoning (WP4)

Traffic-Aware Routing Showcase
Demonstrates Abstraction & Learning (WP3)

Journal Publication
WP6 - Urban Computing Use Case

**Practical**: Involvement of Korean Institute of Building Technology joint with SaltLux. Significant datasets (1B triples) in realistic case-study aimed at Korean govt.

Road Sign Management Demonstrator
Demonstrates Reasoning (WP4)

Traffic-Aware Routing Showcase
Demonstrates Abstraction & Learning (WP3)

Journal Publication
Mímir: Simultaneous Semantic and Text search

{Gene} is implicated in {Disease} of {partOf Lung}
Linking and Mapping Semantics for Early Clinical Development

**Number of Statements** 4,193,400,044

- **Explicit statements** 2,741,704,569
- **Inferred statements** 1,451,695,475

**Cross dataset mappings** 2,156,645

**Semantic Annotations** 263,323,164

**Number of RDF resources** 582,691,283

**Size on disk** ~320GB

**Unique users Jan-Aug 2010** 10,226

---

**Asthma and chronic obstructive pulmonary disease (COPD)** are chronic airway diseases characterized by airflow obstruction. The beta(2)-adrenergic receptor mediates bronchodilation in response to exogenous and endogenous beta-adrenergic agonists. Single nucleotide polymorphisms in the beta(2)-adrenergic receptor gene (ADRB2) cause amino acid changes (e.g., Arg16Gly, Gln27Glu) that potentially alter receptor function.
Example Query

- **Type of data sources**
  - Gene and gene annotations
  - Protein sequences
  - Protein cross references
  - Gene and gene product annotations
  - Organisms
  - Molecular interaction and pathways

- **Database name**
  - Entrez-Gene
  - Uniprot
  - IProClass
  - GeneOntology
  - NCBI Taxonomy
  - BioGRID, NCI, Reactome, BioCarta, KEGG, BioCyc
Example Query

Integrates Multiple Knowledge Sources and Inputs

Give me all proteins which interact in cellular structure and are annotated with repressor and have at least one participant that is encoded by a gene annotated with specific term and is located in chromosome X? Filter the results for Mammalia organisms!
WP 7a Linked Data used through LarKC

• Interest Based Selection (WP2)
  – Refine the results based on who you are
  – Provide a better selection based on who you are

• Random Indexing Selection (WP2)
  – Implement generic RDF search and selection
  – Assist the user in exploring datasets with uncertain schemata

• Causality Chain Mining for Clinical Discovery (WP3)
  – Explore possible causative chains between heterogeneous data sources
  – Rank the results using feedback mechanism
WP 7b: Carcinogenesis Reference Production
New Work, Kidney Cancer

- Initial study (last year) included 650k SNPs genotyped in 1,229 cases and 2,450 controls – study too small and no apparent findings
- Re-ranked SNPs based on LarKC priors - top ranked SNP now rs7579899 (originally ranked 28th)
- Subsequently, study data pooled with multiple GWAS from Europe and US (one year worth of work + $1M) rs7579899 among top 5 ranked SNPs

<table>
<thead>
<tr>
<th>Study</th>
<th>cases</th>
<th>controls</th>
<th>OR</th>
<th>95%CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial study</td>
<td>CE</td>
<td>1096</td>
<td>2062</td>
<td>1.19</td>
</tr>
<tr>
<td></td>
<td>HUNT2/Tromso</td>
<td>133</td>
<td>388</td>
<td>1.07</td>
</tr>
<tr>
<td>Further studies</td>
<td>ASHRAM</td>
<td>84</td>
<td>399</td>
<td>1.32</td>
</tr>
<tr>
<td></td>
<td>EPIC</td>
<td>276</td>
<td>419</td>
<td>1.12</td>
</tr>
<tr>
<td></td>
<td>CeRePP</td>
<td>79</td>
<td>146</td>
<td>1.38</td>
</tr>
<tr>
<td></td>
<td>Moscow</td>
<td>263</td>
<td>306</td>
<td>1.28</td>
</tr>
<tr>
<td></td>
<td>SEARCH</td>
<td>530</td>
<td>1361</td>
<td>1.15</td>
</tr>
<tr>
<td></td>
<td>ATBC</td>
<td>163</td>
<td>1288</td>
<td>1.37</td>
</tr>
<tr>
<td></td>
<td>CPSII</td>
<td>207</td>
<td>730</td>
<td>1.32</td>
</tr>
<tr>
<td></td>
<td>PLCO</td>
<td>284</td>
<td>845</td>
<td>1.05</td>
</tr>
<tr>
<td></td>
<td>USKC</td>
<td>662</td>
<td>561</td>
<td>1.11</td>
</tr>
<tr>
<td>All studies pooled</td>
<td>5936</td>
<td>13,347</td>
<td>1.15</td>
<td>1.10-1.21</td>
</tr>
</tbody>
</table>

- rs7579899 is located in the EPAS1 gene which is involved several mechanisms relevant to kidney cancer
- A report outlining these results has been tentatively accepted for publication in Nature Genetics (Purdue & Johansson et al. 2010)
The LarKC Platform is Powerful & Useful
The LarKC Platform is Powerful & Useful
The LarKC Platform is Available

- open & flexible
- modular & extendable
- distributed & scalable

LarKC@SourceForge
LarKC Codebase
Parallelisation
Plug-In API, Annotations and Plug-in Registry
Remote Execution
Documentation
Data Layer

Realised LarKC Platform Architecture 3.0
LarKC is Real Technology

- open & flexible
- modular & extendable
- distributed & scalable

LarKC@SourceForge
SVN Repository
Maven Libraries
Permissive Licence: Apache 2.0
Support groups
Bug Tracking
Documentation from WPs
鉴于此，由钟宁、刘际明提议并任主编，在高等教育出版社的大力支持下建立了《Web智能与科学》系列丛书，以解决国内Web智能科学研究领域中最新资源不足的状况，给国内研究者提供一个良好的了解和掌握Web智能科学领域的研究动态和研究成果的平台。
Dissemination: Events & Social Media


- LarKC Parallelisation Workshop
  January 31, 2011 - Lyon, France
- LarKC Pharma Workshop
  April 19-20, 2011 - Stuttgart, Germany
- LarKC OpenPhacts Meeting (aka “hackathon”)
  May 25-27, 2011 - Voeren, Belgium

WEB Site: 91000 visits
Blog: 65000 visitors

2100 Views
The Challenge

LarKC Project End

Innovators 2.5%  Early Adopters  Early Majority

us

now

2013

footer
The Challenge

LarKC Project
End

LarKC self
sustaining

us

?  2.5%

Innovators Early Early
now Adopters Majority

2013

now
## Innovation Adoption

### Table: Innovation Adoption Factors

<table>
<thead>
<tr>
<th>Factor</th>
<th>LarKC</th>
<th>How to improve</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative Advantage</td>
<td>High</td>
<td>Standard community-based use driven bug fixes and improvement</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Growing</td>
<td>Semantic web catching on; plugins for popular web services</td>
</tr>
<tr>
<td>Complexity / Simplicity</td>
<td>Complex</td>
<td>e.g. workflow designer, but inherent complexity; mitigate with Trialability</td>
</tr>
<tr>
<td>Trialability</td>
<td>Low</td>
<td>Need to get from 15 min to run to 30 min to modify interesting example and run</td>
</tr>
<tr>
<td>Observability</td>
<td>Low</td>
<td>Start with single community e.g. pharma</td>
</tr>
</tbody>
</table>

---

Ready to Adopt

• The platform is useable
• Use-cases, with major effort, have driven platform and demonstrated feasibility

• It’s ready to transition to a user community
“Create an **open** knowledge infrastructure enabling facile **integration** of chemical and biological data to support **drug discovery**”

Data sources
- Linked Open DrugData
- ChEMBL
- Spider
- Linked LifeData
- Wiki Pathways

Identity Resolution Service
- Concept WikiUI
- Concept Wiki
- Term mapping
- Bridge DB
- URI mapping

User Interface software
- Utopia Docs
- Path Physio

IRS to disambiguate

SPARQL

LarKC

LarKC in the core

SPARQL or LarKC plugin

Distributed system

Companies involved:
- Pfizer limited
- Universität Wien
- Technical University of Denmark
- University of Hamburg, Center for Bioinformatics
- BioSolveIT GmbH
- Consorci Mar Parc de Salut de Barcelona
- Leiden University Medical Centre
- Royal Society of Chemistry
- Vrije Universiteit Amsterdam
- Spanish National Cancer Research Centre
- University of Manchester
- University of Maastricht
- ACKnowledge
- University of Santiago de Compostela
- Rheinische Friedrich-Wilhelms-Universität Bonn
- AstraZeneca
- GlaxoSmithKline
- Laboratorios del Dr. Esteve, S.A.
- Novartis
- Merck Serono
- H. Lundbeck A/S
- Eli Lilly

OpenPhacts
12m€ budget, 3 YR

funded
“Create an open knowledge infrastructure enabling facile integration of chemical and biological data to support drug discovery”

“LarKC considered as the only chance to build a prototype in 6 months”

“The thing about LarKC is that the pipeline approach makes it easy to test out new infrastructure pipelines. For example, integrating identity resolution and chemistry services into those pipelines.”

OpenPhacts
12m€ budget, 3 YR

Pfizer limited
Universität Wien
Technical University of Denmark
University of Hamburg, Center for Bioinformatics
BioSolveIT GmbH
Consorti Mar Parc de Salut de Barcelona
Leiden University Medical Centre
Reial Society of Chemistry
Erasmus Universiteit Amsterdam
Spanish National Cancer Research Centre
University of Manchester
University of Maastricht
University of Santiago de Compostela
Rheinische Friedrich-Wilhelms-Universität Bonn

AstraZeneca
GlaxoSmithKline
Laboratorios del Dr. Esteve, S.A.
Novartis
Merck Serono
H. Lundbeck A/S
Eli Lilly
Cycorp (US) NLM Causal Pathways with LLD/Owlim
Cycorp (US) NLM Causal Pathways with LLD/Owlim

Rule

In Mt: MolecularBiologyMt
(implies
(isa ?MOLECULE-TYPE TranscriptionFactor)
(behaviorCapable ?MOLECULE-TYPE
(ChemicalBindingEventTypeWithTypesFn TranscriptionFactor DNAMolecule)
(objectOfAttachment)))

Transcription factors can bind with DNA.

NSAIDs Use Case Astrazeneca

Detailed Justification:

- There is some evidence that ETS2 affects NCOA3.
- There is some evidence that BMP7 affects positive regulation of osteoblast differentiation.
  - If the evidence given for a type of biological process having a type of chemical object as a participant is experimental evidence (as opposed to evidence based on structural similarity or computer matching), and a particular gene codes for that chemical object type, then conclude that instances of the process type typically have the product of that gene as participants.
  - Inferred from direct assay is a type of experimental evidence.¹
- The evidence for Bone morphogenetic protein 7's participation in the positive regulation of osteoblast differentiation was provided by direct assay.
- BMP7 codes for Bone morphogenetic protein 7.
  - If, according to EntrezGene, a gene has a particular term in UniProt as its "uniprotAccession", then gene codes for the type of peptide molecule represented by the UniProt term.
  - According to EntrezGene, "BMP7" codes for Bone morphogenetic protein 7.

- There is some evidence that NCOA3 affects BMP7.
  - Inferred from direct assay is a type of experimental evidence.¹
- There is some evidence that positive regulation of osteoblast differentiation affects the quantity of bone tissue present.
- Positive regulation of osteoblast differentiation is more specific than biological processes.

External Sources:

1. Linked Life Ontology
Future of LarKC

MSEE – Manufacturing SErvice Ecosystem
Large-scale Integrating Project (2011-2014) (Objective: 7.3)

VISION
By 2015, novel service-oriented management methodologies and the Future Internet universal business infrastructure will enable European virtual factories and enterprises to self-organize in distributed, autonomous, interoperable, non-hierarchical innovation ecosystems of tangible and intangible manufacturing assets, to be virtually described, on-the-fly composed and dynamically delivered as a Service, end-to-end along the globalised value chain.

Contribution of LarKC:
- Adoption of its architecture to meet the needs of a scalable architecture that supports data and services consumption on a large scale.
• envision – Environmental service infrastructure with ontologies.

**OBJECTIVE**

Build an **ENVironmental Services Infrastructure with ONtologies** that aims to support non ICT-skilled users in the process of **semantic discovery and adaptive chaining and composition** of environmental services.

• Contribution of LarKC:
  – *Reasoning over sensor data streams in different environmental scenarios (oil spill predictions, flood risk management and landslide predictions)*
• Use user interests for knowledge retrieval and during reasoning

**Usecases**

**Usecase 1** The large scale scientific literature semantic search and analysis system as well as the active academic collaboration and visit recommendation system to be developed under this project will provide many useful supporting functionalities to assist researchers in conducting scientific research. In addition, the developed system is aimed at supporting the research on the Science of Team Science [2].

Now the very preliminary test version of the sub system "Active Academic Visit Recommendation Application (AAVRA)" is in prototype shape. It is based on the LarKC platform, with 3 workflows (including 7 plugins running within these workflows).
Future GWAS exploitation will include:

- Publish the underlying approach and evaluation
- Large savings for the cancer research community
- Hopefully, help find more causes for cancer

Applying the same approach to any study investigating a large number of genes:

- 2nd generation studies
- Exome sequencing
- Epigenetics
• Complex, large data inference over complex models
• Rough plan: Two interacting use cases:
  1. Deep semantic NLU by inference alone, over large corpora;
  2. NLM-like, with deep biology
• Complex, large data inference over complex models
• Rough plan: Two interacting use cases:
  1. Deep semantic NLU by inference alone, over large corpora;
  2. NLM-like, with deep biology
• Complex, large data inference over complex models
• Rough plan: Two interacting use cases:
  1. Deep semantic NLU by inference alone, over large corpora;
  2. NLM-like, with deep biology

Induces cartilage and bone formation. Also act in mesoderm induction, tooth development, limb formation and fracture repair. Acts in concert with PTHLH/PTHRP to stimulate ductal outgrowth during embryonic mammary development and to inhibit hair follicle induction (By similarity).

E: The presence of Bone morphogenetic protein 4 in mammal causally contributes to cartilage development and the presence of Bone morphogenetic protein 4 in mammal causally contributes to bone formation.
• The Linked Data Quizzes project aims at leveraging the Linked Data through a natural language interface and the automatic generation of quizzes.
• To provide the starting point for creating quizzes the central task is that of clustering the Linked Data cloud into such subgraphs that correspond to various topics of interest (e.g., music, history, etc.).
• Each cluster will serve as high-level selector of the desired area of quiz questions.
• For this task we will reuse the methodology for virtual documents and the selection methods and plugins from LarKC (random indexing, spreading activation and baseline).
Life Cycle Analysis
Knowledge and Data
• Semanticising ArXiv.org (500,000 documents)
• physics ontology made by physicists
• semantic facetted navigation
• crowd-sourced bookmarking
• homegrown implementation in MySQL
• now migrating to SKOS & LarKC
• they do all the migrating and programming
• we do handholding and consulting
• based on response from Mark Musen, member of our industrial advisory board
• Protégé focuses on the /data/, but not on what to do with the data
• LarKC has a strong notion of workflow & plugins that is nicely complementary with Protégé (because different functionality) but also compatible (because same data model)
• commitment from Protégé team to work with LarKC on integrating the LarKC and Protégé
• gives LarKC access to large and active Protégé community
LarKC as a backend app server

It is curious about:

- **You**: your life, and your needs
- **The World**: in detail
- **The Connections**: it wants to make your life better, and tries to learn how

<table>
<thead>
<tr>
<th>Foursquare</th>
<th>Google</th>
<th>Facebook</th>
<th>CYC</th>
</tr>
</thead>
<tbody>
<tr>
<td>LarKC Platform</td>
<td>+</td>
<td>CYC Platform</td>
<td></td>
</tr>
<tr>
<td>SPARQL ENDPOINT</td>
<td>CYC NL ENDPOINT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integration server</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Bottari Mobile App
- City of Seoul and Saltlux have agreed official distribution and common marketing of Bottari app
- Commercial services will be available in other countries: Japan (‘12), China (‘13) and Europe (‘14)

Road Sign Management
- Government organization KICT already introduced LarKC and RSM application for city management
- KICT has a plan to utilize LarKC and RSM for designing and maintaining new capital city of Korea

Traffic Prediction
- Aggressive disseminating and PR of Traffic LarKC to government sectors
- Siemens and Saltlux will propose Traffic LarKC to car navigation companies for commercial services
Slovenian Company, Light-as-a-service: Economics based on sensing and reasoning about light needs, LarKC to process, integrate, sensor data.
Committed to the Future of LarKC

• Commitment from partners (+ JSI) for continuing support of released code
  – UIBK
  – VUA
  – JSI
  – WICI
  – SoftGress

http://wiki.larkc.eu/KeepingLarKCAlive
How we’re planning to make LarKC thrive

• Setting up STI Working Group with goal to:
  – Extend Open Source infrastructure to include all project outputs and deliverables
  – Build and sustain Open Source community around platform and its users, as take-up accelerates
  – Transition responsibility for LarKC into that community

• Improve Market Place and integration with the platform
  • Integration of end-user frontends
  • Training materials, documentation
LarKC is the right technology
Platforms take time
Sustained support
Sustained use