Stratified Composition of Web Services[1]

Riina Maigre

Institute of Cybernetics at TUT

Theory Days at Kääriku 2009

---

[1]Based on: R. Maigre, P. Grigorenko, P. Küngas , E. Tyugu. Stratified Composition of Web Services. In: M. Virvou, T. Nakamura (eds.) Knowledge-Based Software Engineering. Proc. 8th JCKBSE. IOS Press, 2008, p. 49 - 58

Outline

Motivation

Large information systems with service oriented architecture

Knowledge architecture of the composition tool

User knowledge level

Logical level

Service implementation level

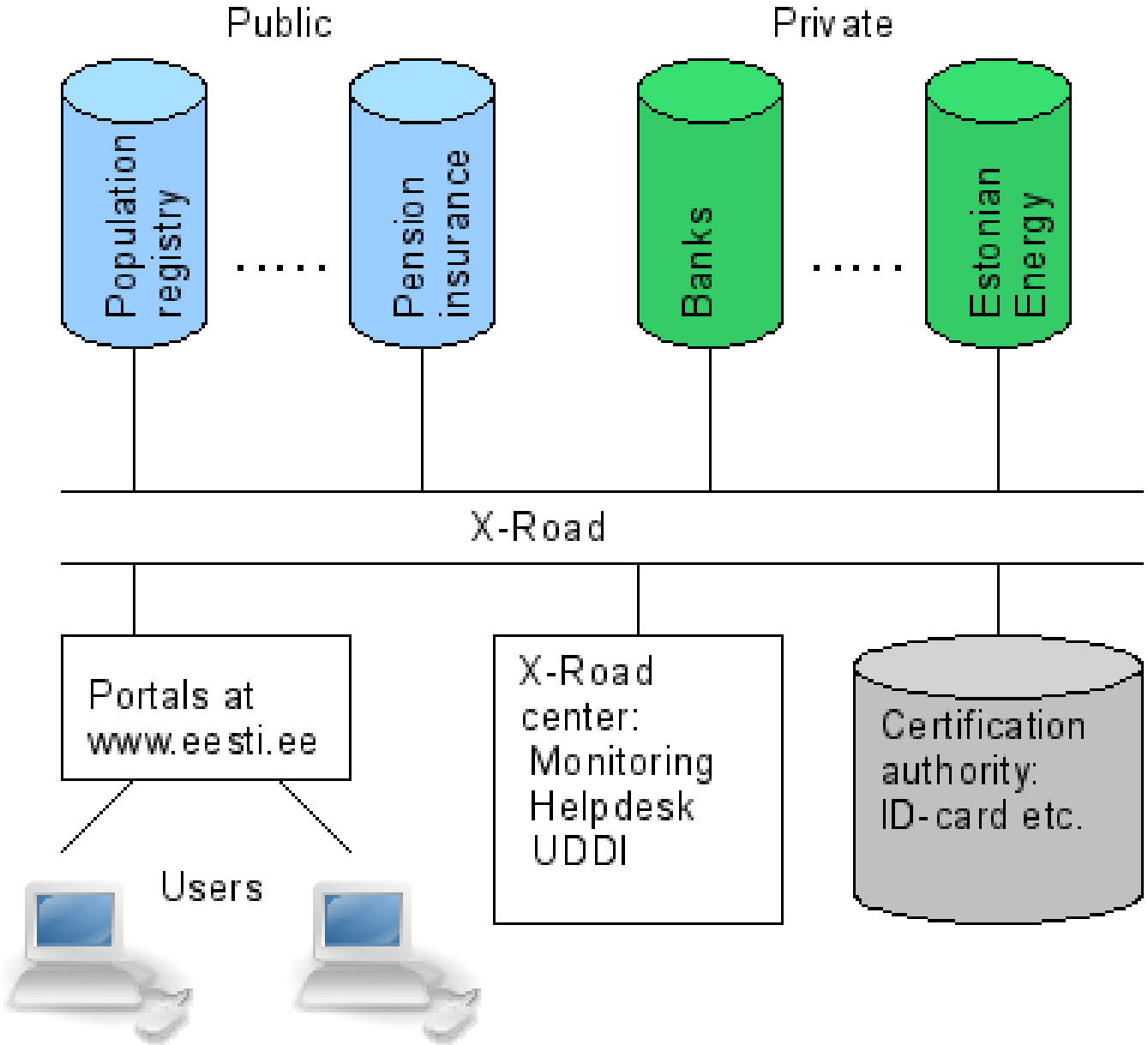
Motivation

- Goal of this work is to overcome the complexity of service composition on very large sets of atomic services.
- To achieve this goal we propose an automatic service composition methodology with three distinguished knowledge levels:
  - user knowledge level
  - logical level
  - implementation level

Example application domain

- Estonian e-government information system has a service-oriented architecture.
  - Services accessible through this information system are described in WSDL.
  - More than thousands of services are provided by information systems of different institutions.
- Secure access is provided (over X-Road) to nearly all public (i.e. governmental) but also some private databases.
- All Estonian residents having the national ID card can access these services through X-Road within their limits of authority.
- In this work we refer to the whole system as X-Road.

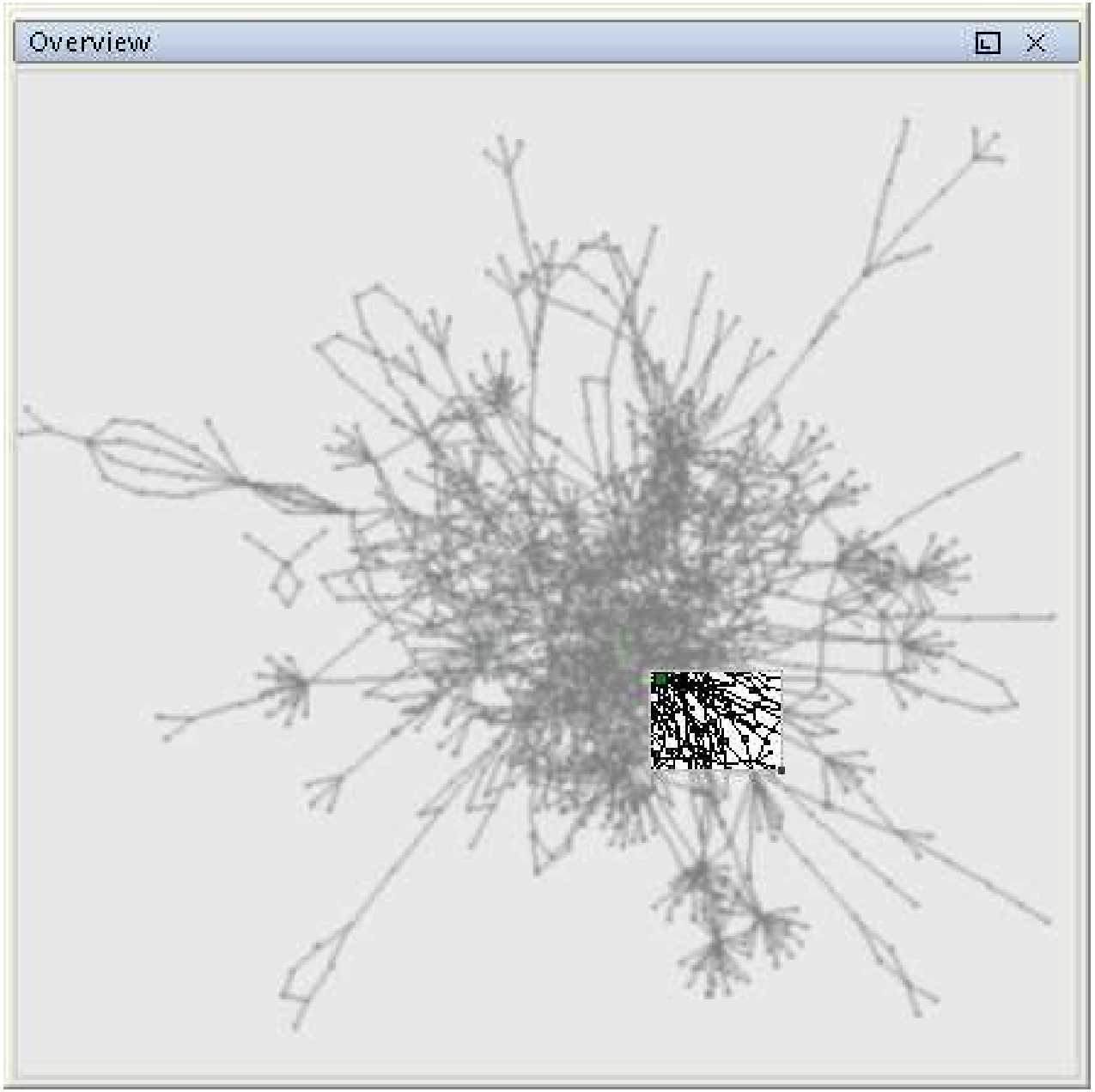
X-Road

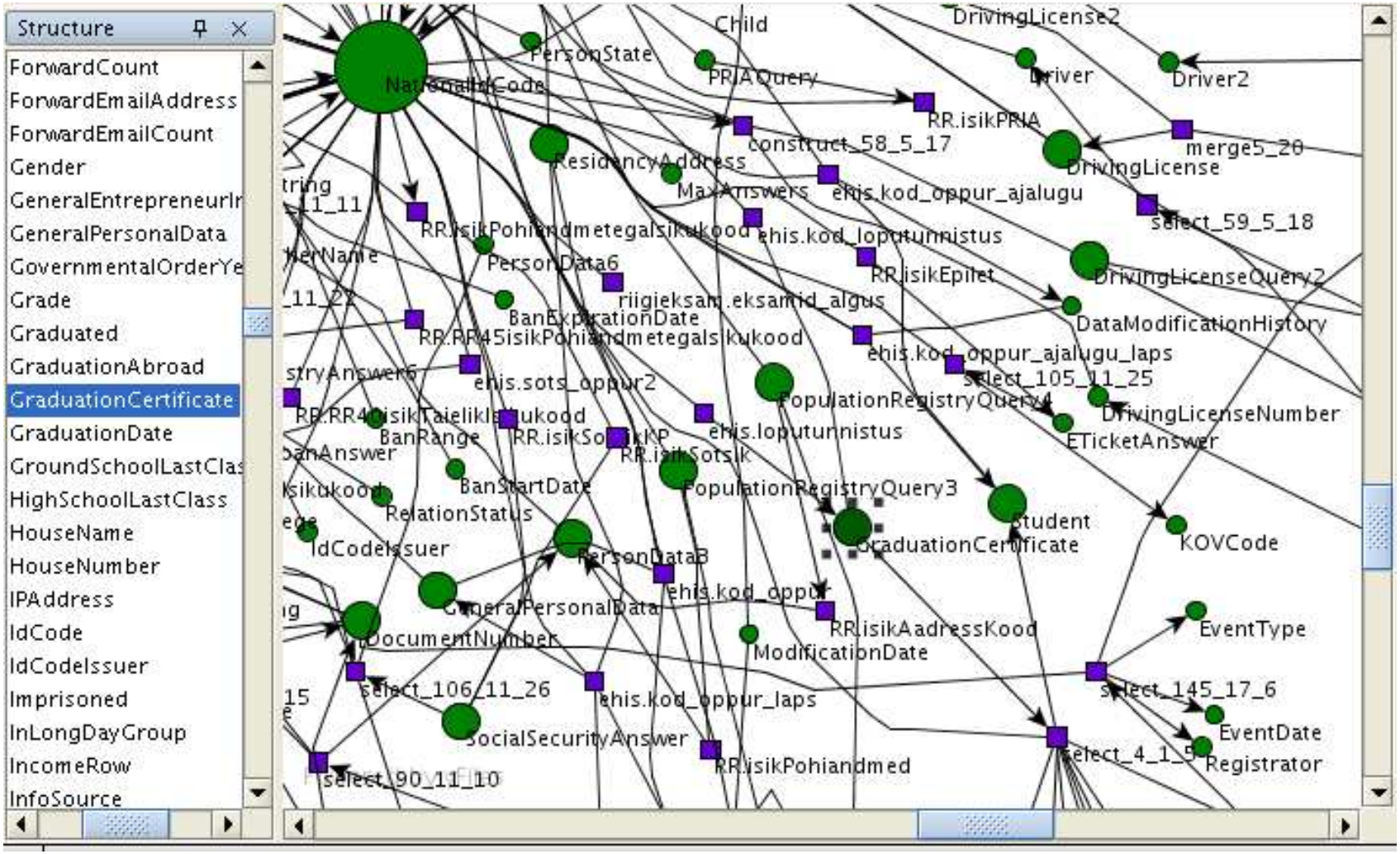

Public
Private
Population registry
Pension insurance
Banks
Estonian Energy
X-Road
Portals at www.eesti.ee
X-Road center: Monitoring Helpdesk UDDI
Certification authority: ID-card etc.
Users

Example model

- We have performed experiments on a part of X-road information system.
- Service model containing about 300 atomic services and about 600 references to semantic resources has been created.
- Totally more then thousand of atomic services are available.

Model of the application domain




Structure
ForwardCount
ForwardEmailAddress
ForwardEmailCount
Gender
GeneralEntrepreneurIr
GeneralPersonalData
GovernmentalOrderYe
Grade
Graduated
GraduationAbroad
GraduationCertificate
GraduationDate
GroundSchoolLastClas
HighSchoolLastClass
HouseName
HouseNumber
IPAddress
IdCode
IdCodeIssuer
Imprisoned
InLongDayGroup
IncomeRow
InfoSource
Child
DrivingLicense2
PersonState
PRIAQuery
Driver
Driver2
RR.isikPRIA
construct_58_5_17
merge5_20
DrivingLicense
ResidencyAddress
MaxAnswers
ehis.kod_oppur_ajalugu
select_59_5_18
RR.isikPohiandmetegalsikukood
ehis.kod_loputunnistus
PersonData6
RR.isikEpilet
DrivingLicenseQuery2
BanExpirationDate
DataModificationHistory
RR.RR45isikPohiandmetegalsikukood
ehis.kod_oppur_ajalugu_laps
ehis.sots_oppur2
select_105_11_25
PopulationRegistryQuery4
DrivingLicenseNumber
ETicketAnswer
ehis.loputunnistus
BanRange
RR.isikSotsik
BanStartDate
PopulationRegistryQuery3
Student
RelationStatus
GraduationCertificate
KOVCode
IdCodeIssuer
PersonData3
ehis.kod_oppur
GeneralPersonalData
RR.isikAadressKood
EventType
DocumentNumber
ModificationDate
select_106_11_26
ehis.kod_oppur_laps
select_145_17_6
EventDate
SocialSecurityAnswer
select_4_1_5
Registrator
RR.isikPohiandmed
select_90_11_10

Querying X-Road

- Only predefined queries can be done through X-Road portals
- Combined queries between information systems of different institutions are not possible.
- Possible semantic relationships between queries are ignored.

Example: quering person's contact addresses

(Without complex queries:)

From the 1th DB: *NationalIdCode* $\rightarrow$ *Address*
From the 2nd DB: *NationalIdCode* $\rightarrow$ *Address*
...
From the n-th DB: *NationalIdCode* $\rightarrow$ *Address*

Example complex query

$$\mathit{NationalIdCode} \rightarrow$$

$$\mathit{AddressString}, \mathit{EstonianAddressString},$$

$$\mathit{OwnerAddressString}, \mathit{ResidencyAddress},$$

$$\mathit{ResponsibleUserAddress}$$

Service composition tool

- We have created a tool for service composition that supports the stratified composition methodology and hides complexity of composition from the end user.
- Tool is created in the software development environment CoCoViLa that supports automatic synthesis of programs and generates Java code from visual and textual specifications.

Knowledge architecture of the composition tool

Knowledge system – module of knowledge architecture.

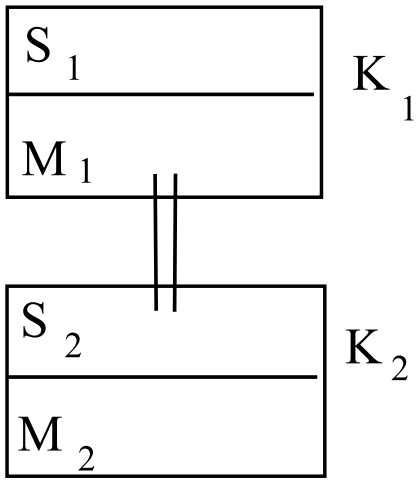
- knowledge language
- knowledge handling mechanism
- method for associating meanings to knowledge objects

S – knowledge objects (notations)
M – set of meanings (denotations)

Connections of knowledge systems

Knowledge systems can be composed into larger knowledge architecture by using:

- Hierarchical connections (a)
- Semantic connections (b)
- Operational connections



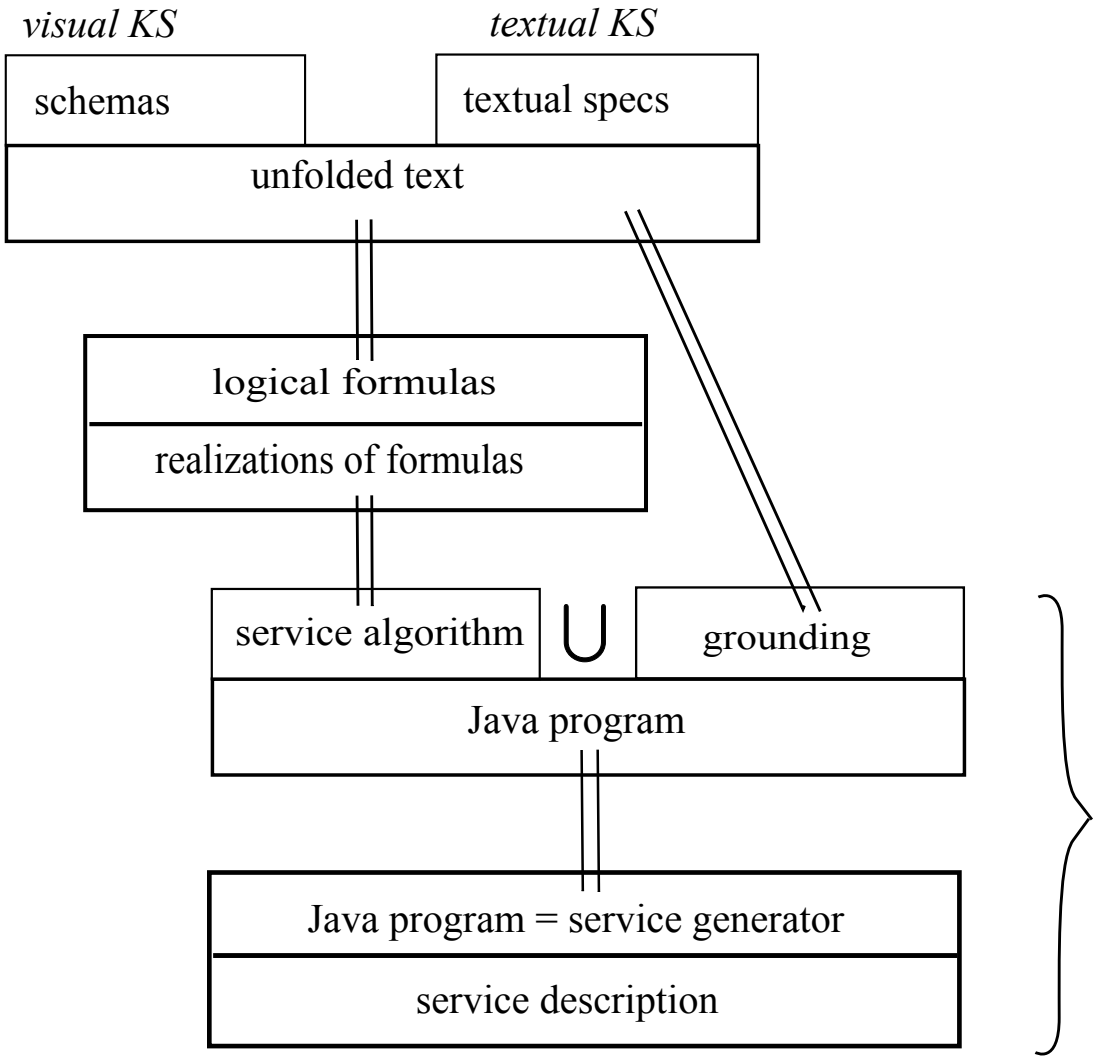
a)

b)

Knowledge levels

1. User knowledge level
2. Logical level
3. Service implementation level

Service composition tool

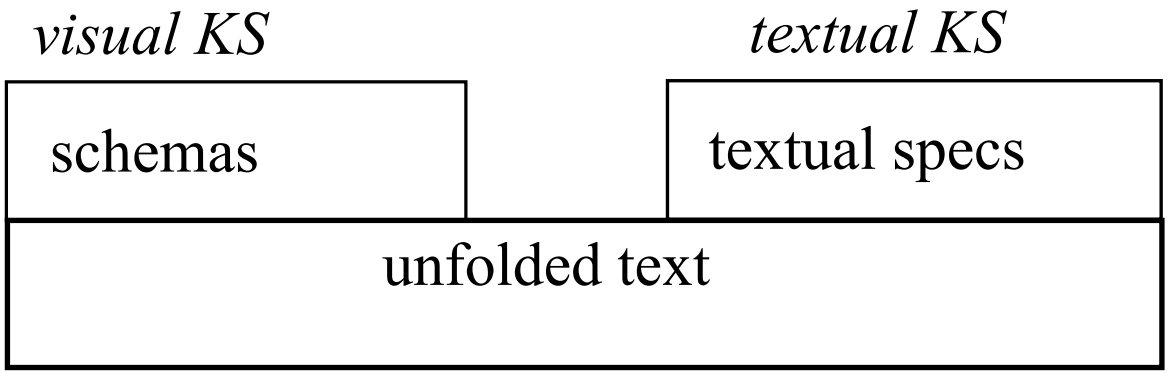


user knowledge level

logical level
(intuitionistic logic)

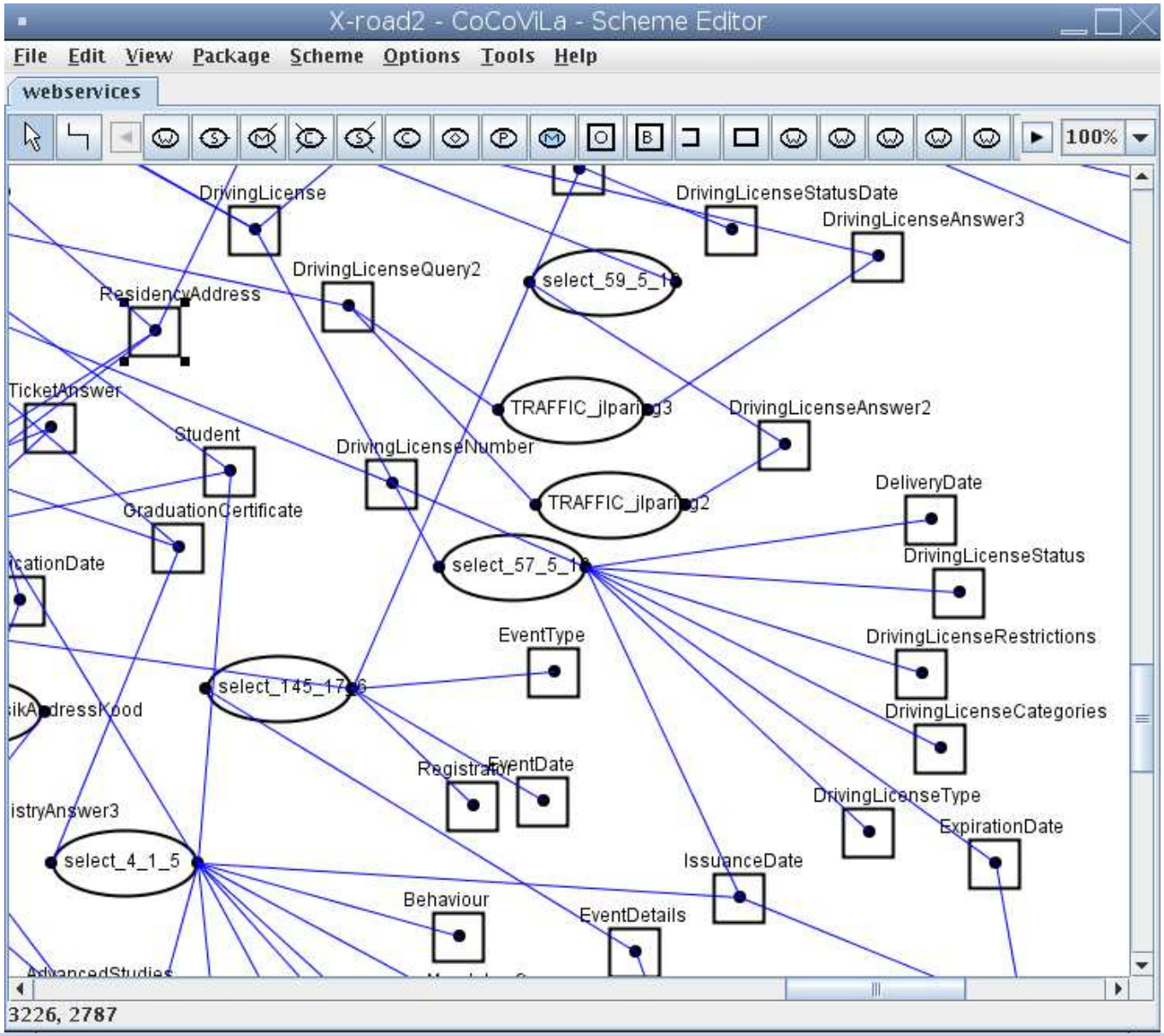
service implementation level

1. User knowledge level



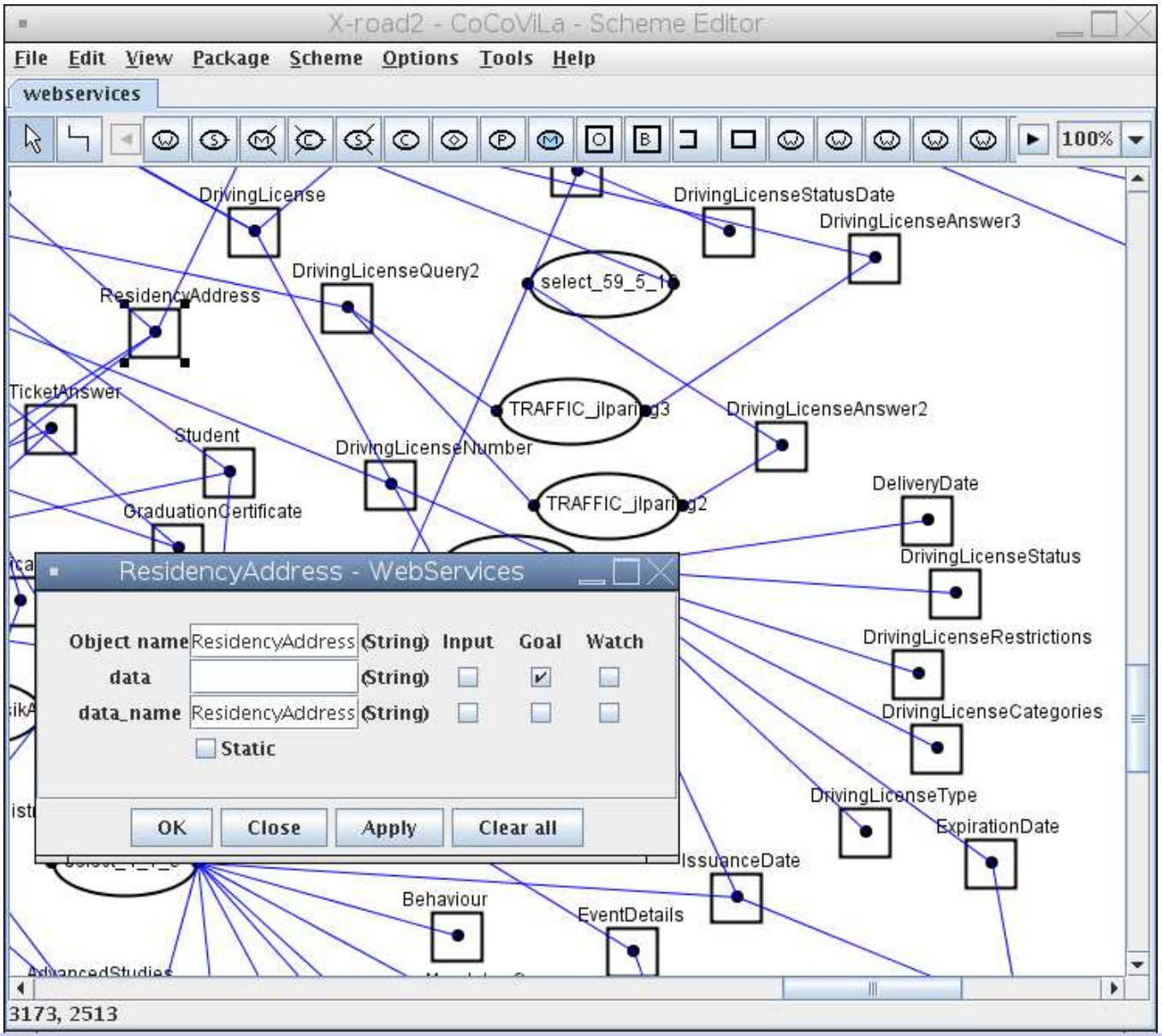
1. User knowledge level

Visual knowledge representation.



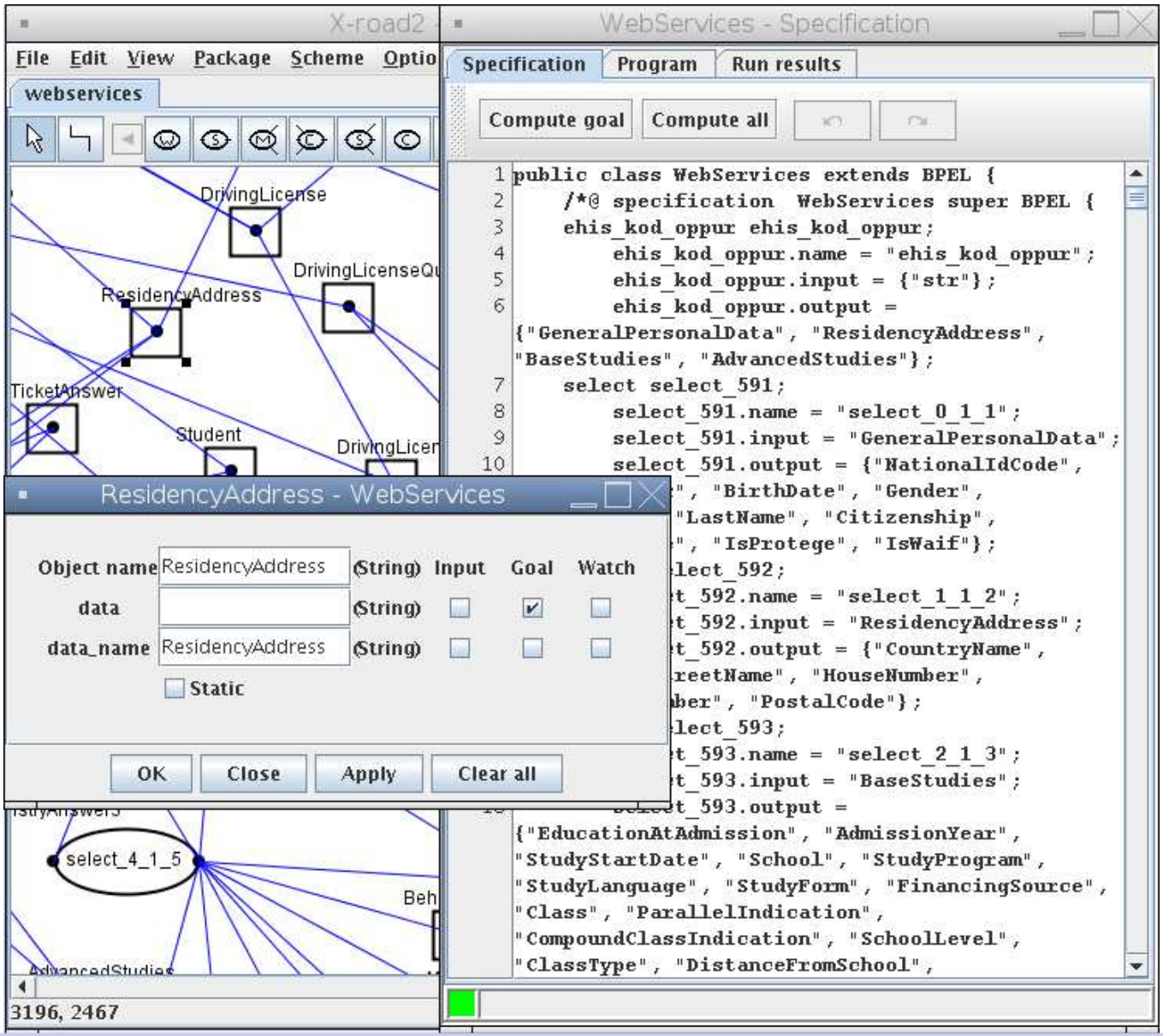
1. User knowledge level

Specifying a complex service on visual representation.



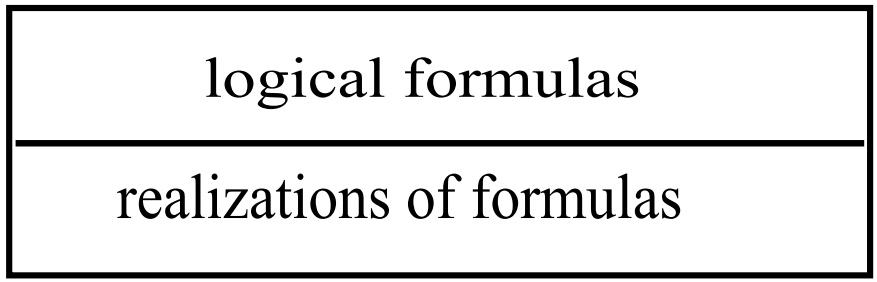
1. User knowledge level

Visual and textual representation of knowledge.



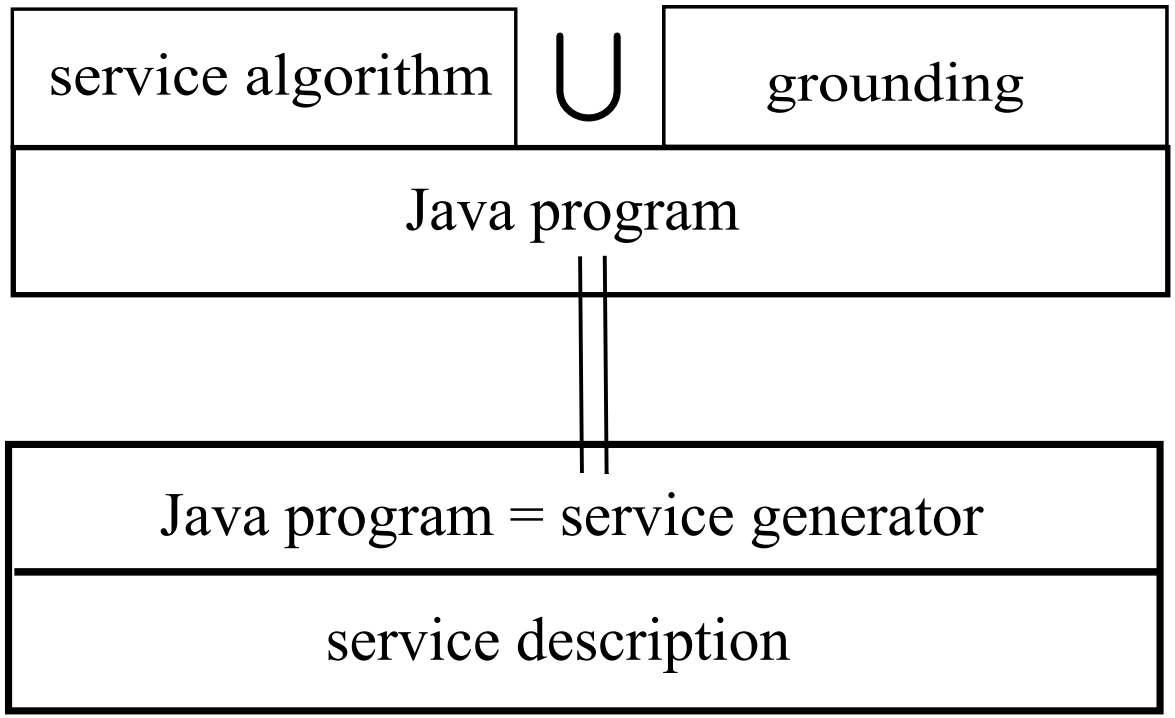
2. Logical level

Formal representation of knowledge and automatic composition of new services.



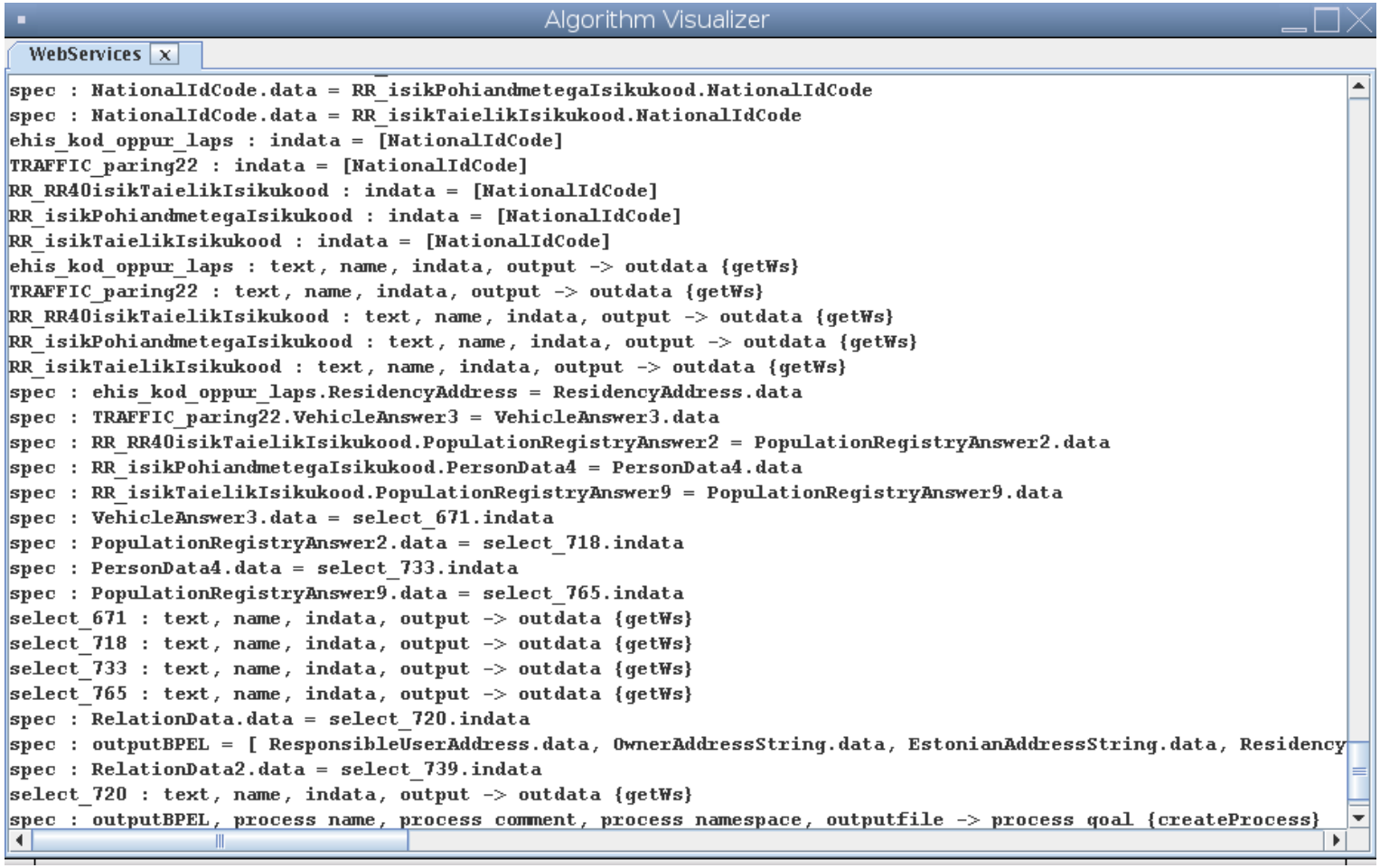
- Knowledge about atomic services, goal => algorithm of expected service
- Structural synthesis of programs is used to synthesize a structure of a new complex service

3. Service implementation level



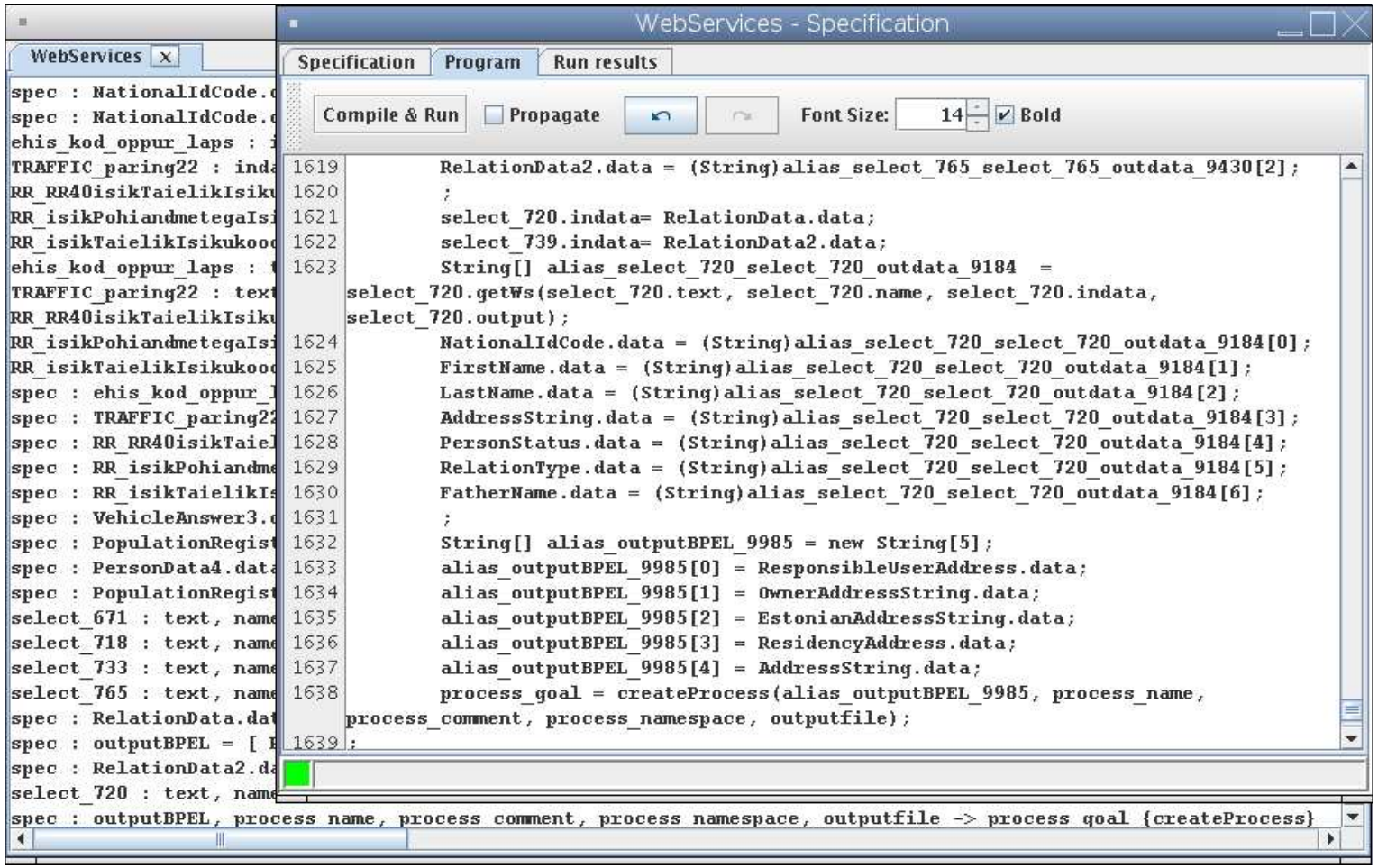
3. Service implementation level

Synthesizing a structure of the complex service.



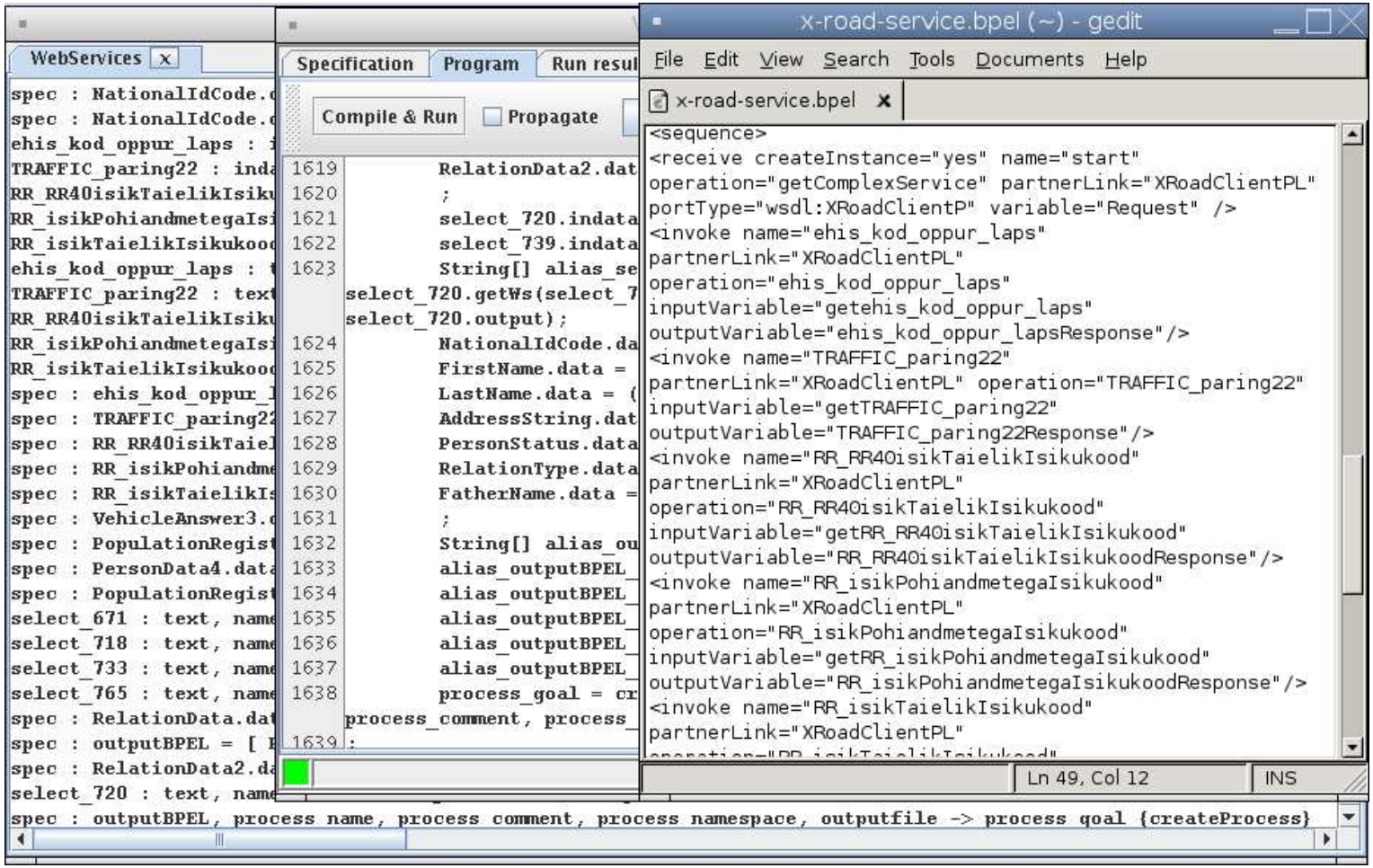
3. Service implementation level

Grounding of services - calling services and performing actual computations.



3. Service implementation level

Generating BPEL from Java code.



Conclusion

- We have described the architecture of the complex knowledge-based tool with stratified knowledge levels.
- The main features of the presented architecture are
  - user friendly upper level
  - precisely defined mappings between levels
- We hope to make the tool usable first of all to the developers

Thank you for listening!

Questions?

Supporters:
Estonian Science Foundation

Tiger University