A framework to provide the distributed execution of semantic and image analysis algorithms on different platforms and architectures

M. FLÜGGE, N. ZERBE, K. SCHLÜNS, P. HUFNAGL

INSTITUTE OF PATHOLOGY
CHARITÉ, UNIVERSITY HOSPITAL BERLIN

09.07.2010
Virtual Specimenscout Project

- Achieves to provided a **diagnostic platform** for histological analysis on whole slide images
- Automatic **pre-analysis** of slides
- On **demand analysis** capabilities combined with virtual microscopes
- **Search engine** for diagnostic results which allows textual and image based requests and presents comparable **medical cases**
Motivation

• Need for cross-platform architecture because …
  o Different image analysis frameworks, tools and APIs are available (Definiens Developer, ImageJ, ITK,…)
  o Different clients (e.g. viewers) need to be integrated (written in different languages)
  o Different levels of skills can be problematic if only one platform is used
  o Wrong decision might cost a lot of money and time during projects

• Conclusion
  o We need a system which provides (distributed) platforms that encapsulate the framework specific algorithms
  o S⁴ – Specimen Scout Service System
Algorithm Execution Platform

- Algorithm Processing Environment - APE
  - Every algorithm will be integrated in an Algorithm Processing Platform
  - Is a converter between particular platform and the $S^4$ interfaces
- Two types
  - Image Processing Environment - IPE
  - Semantic Processing Environment - SPE
Algorithms

- **Elementary Algorithms**
  - The primitive algorithms provided by the particular platform
  - Written in platform specific language

- **Basic Algorithms**
  - Are encapsulating Elementary Algorithms
  - Translating the Elementary Algorithms and the core system
  - Written in IPE language

- **Compound Algorithms**
  - Consists of Basic Algorithms and/or Compound Algorithms
Communication Infrastructure

Client

Slidy
(Java)

Slide Explorer
(C#)

Definies IPE
(SOAP)

ITK IPE
(C/C++)

ImageJ IPE
(Java)

VMScope IPE
(C#)

IPE

FrontController

requests algorithm execution

returns result

requests algorithm execution

returns result
Communication connectors

- Connectors hide the access to the system
- Allows to change the communication protocol
- Even possible to switch connection at runtime
- Currently implemented
  - SOAP based access
    - Distributed and flexible
    - Generators for many languages available
  - Memory access
    - Locally on the server or client
    - Higher performance
Summary

- S⁴ provides
  - The platform provides the potential to **simplify development** of image analysis algorithm
  - Especially useful for **prototyping** because algorithms can be exchanged
  - Use of common **distributed** technologies (e.g. SOAP) makes access quite easy
  - Flexible **connectors** allow multiple scenarios (e.g. fully distributed)
Outlook

- Integrate a bunch of useful algorithms
- Provide access to semantic frameworks
- Provide user interfaces for development (e.g. graphical editors)
- Provide own simple language to access the frameworks capabilities

MyScript
{
    Image image = ContentProvider.get(“myNiceImage“);
    Image result = on ImageJ do GaussianFilter(image, 0.8)
    Image result2 = on Defininens do Classification(result);
    ...
}
Thank you for your attention!