

Data and Application Integration in Learning Content Management Systems: a Web Services Approach

Ivan Madjarov & Omar Boucelma

EC-TEL 2006

Crete, Greece - October 1-4, 2006

Summary

- e-Learning Systems
 - Status
 - Limitation
- Proposed solutions
- Service-based e-Learning environment
 - Course Semantic Structure and Tools
 - LMS and LCMS
 - Service-Oriented System Architecture
 - Use case scenario
- Conclusion

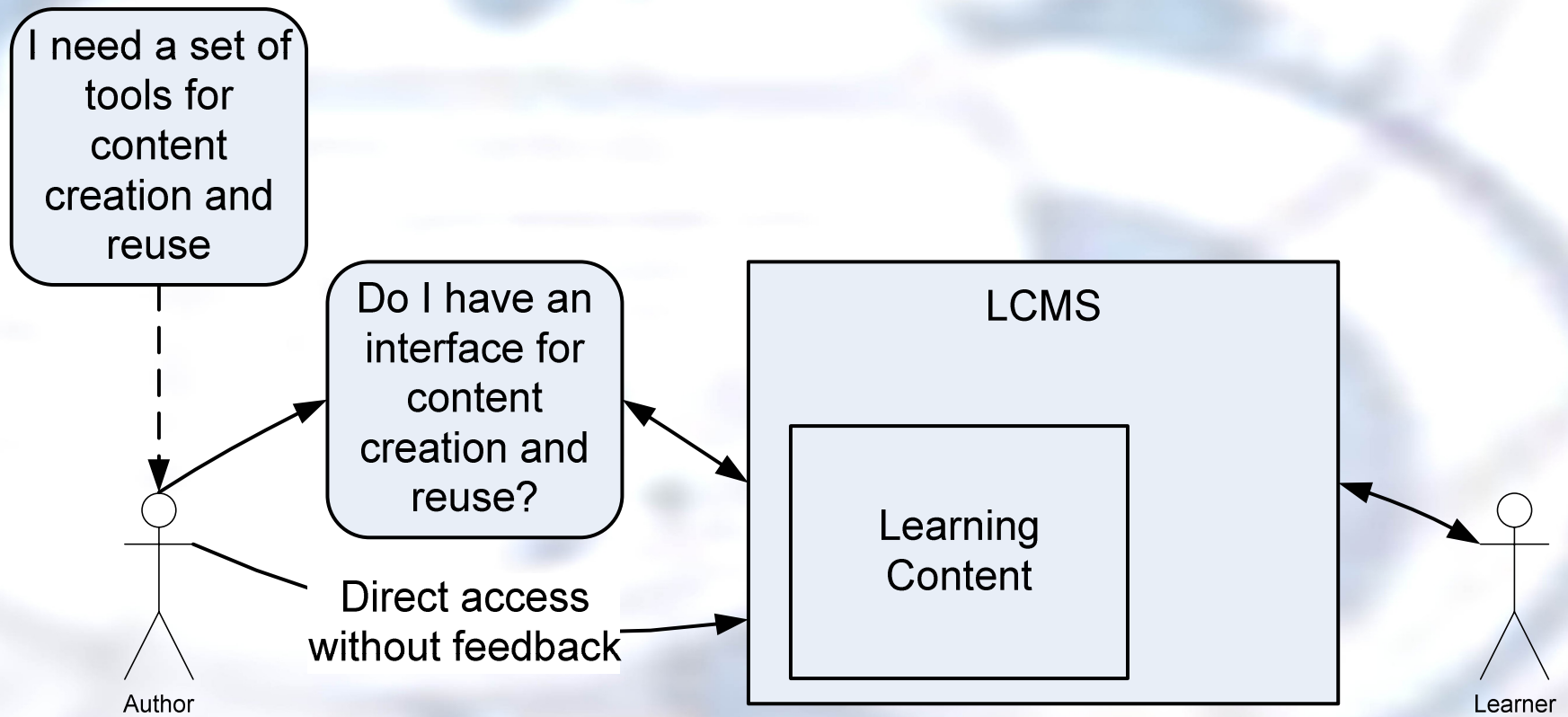
e-Learning Systems: Status

- e-Learning systems are based on models :
 - plain monolithic,
 - component client-server,
 - peer-to-peer.
- e-Learning interoperability system is based on standards :
 - LOM (*Learning Object Metadata*),
 - SCORM (*Sharable Content Object Reference Model*),
 - IMS Global Learning Consortium.

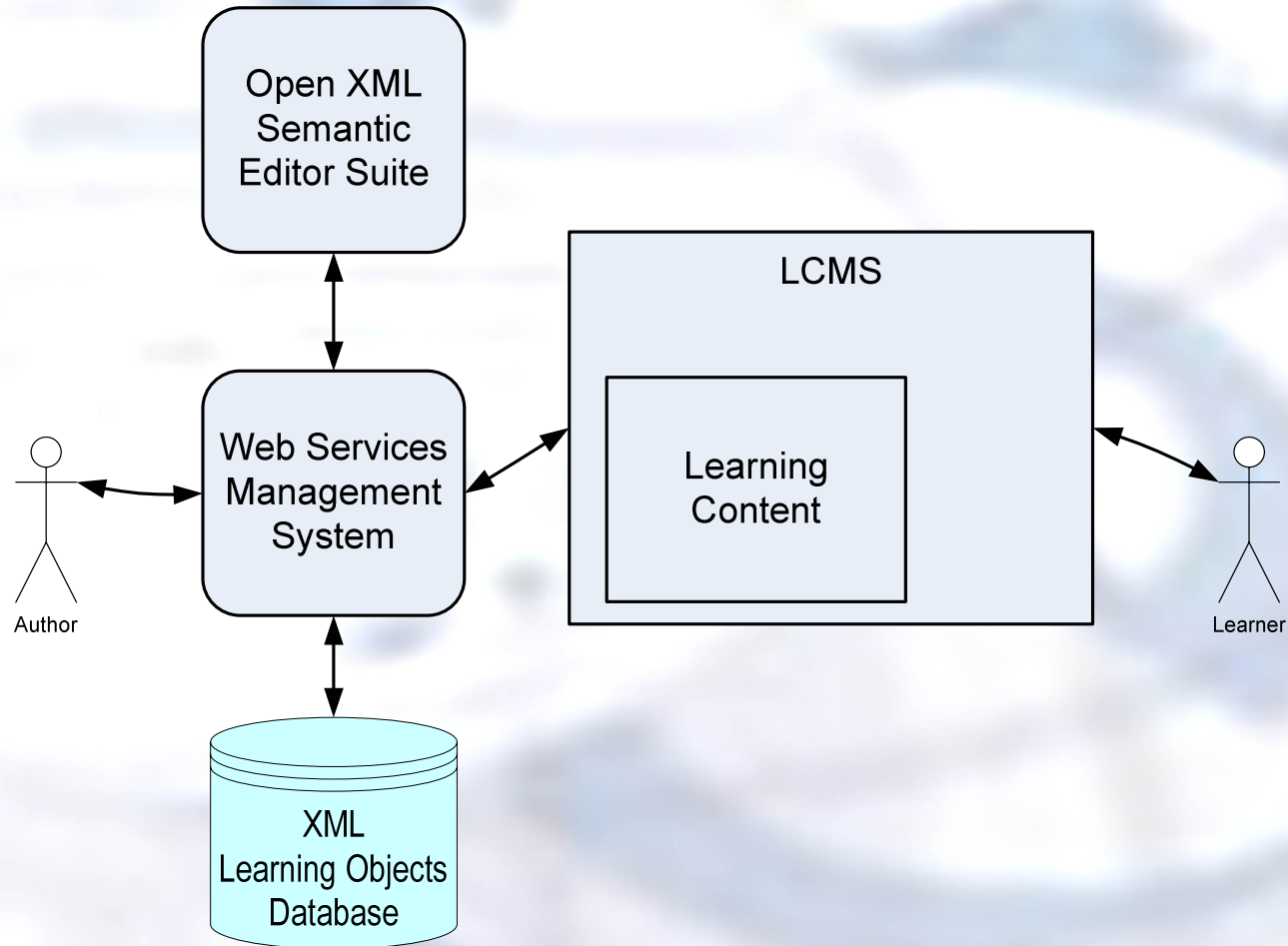
e-Learning Systems : Limitation

- Systems suffer from drawbacks like :
 - poor scalability,
 - low availability.
- Complicated interchange of content
- Standard specification at various levels of granularity
- Application integration is based on API's
 - Replacing or updating any software component means reworking of the API's

The problem



Problem solution



Our Solution (1)

- Interoperability and reusability for learning data and applications based on :
 - Service oriented architecture (SOA) for LCMS,
 - Web services for interconnections between heterogeneous systems,
 - AJAX service-based technique for Web-based requests.

Our Solution (2)

- Integration for learning data and applications based on :
 - open XML semantic and Web service-based authoring suite :
 - environment to create, edit and store LOs;
 - publish e-Learning content in a LCMS content database.
- e-Learning system should consist of :
 - Independent but cooperating services-based applications,
 - pedagogical data integration between common LCMS.

Our Solution (3)

- The role of SOA we suggest is to :
 - Encapsulate the pedagogical content inside a Web service in order to :
 - increase system interoperability,
 - increase learning data and applications reusability.
 - Integrate existing free LCMS with different external components as Web services.
 - Assembly distributed services to provide the functionality that really need.

e-Learning environment

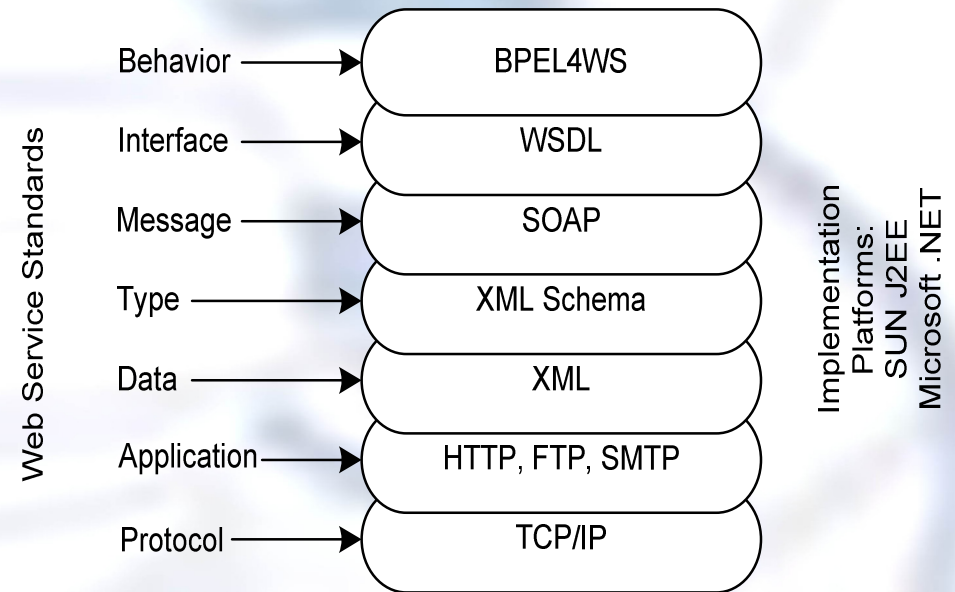
- Groups of people involved :
 - authors and learners;
 - administrators and trainers.
- The core of e-Learning system :
 - LMS (*Learning Management System*) :
 - management of learners and their profiles,
 - progress tracking, collaboration facilitation.
 - LCMS (*Learning Content Management System*) :
 - eases content reusability,
 - workflow support for content development,
 - content delivering via predefined interfaces.

Service Oriented Architecture

- Services are components
 - Well defined interfaces
 - Implementation independent
- Self-contained and loosely coupled
- Composite services can be assembled from individual services
- Using WS and SOAP over HTTP

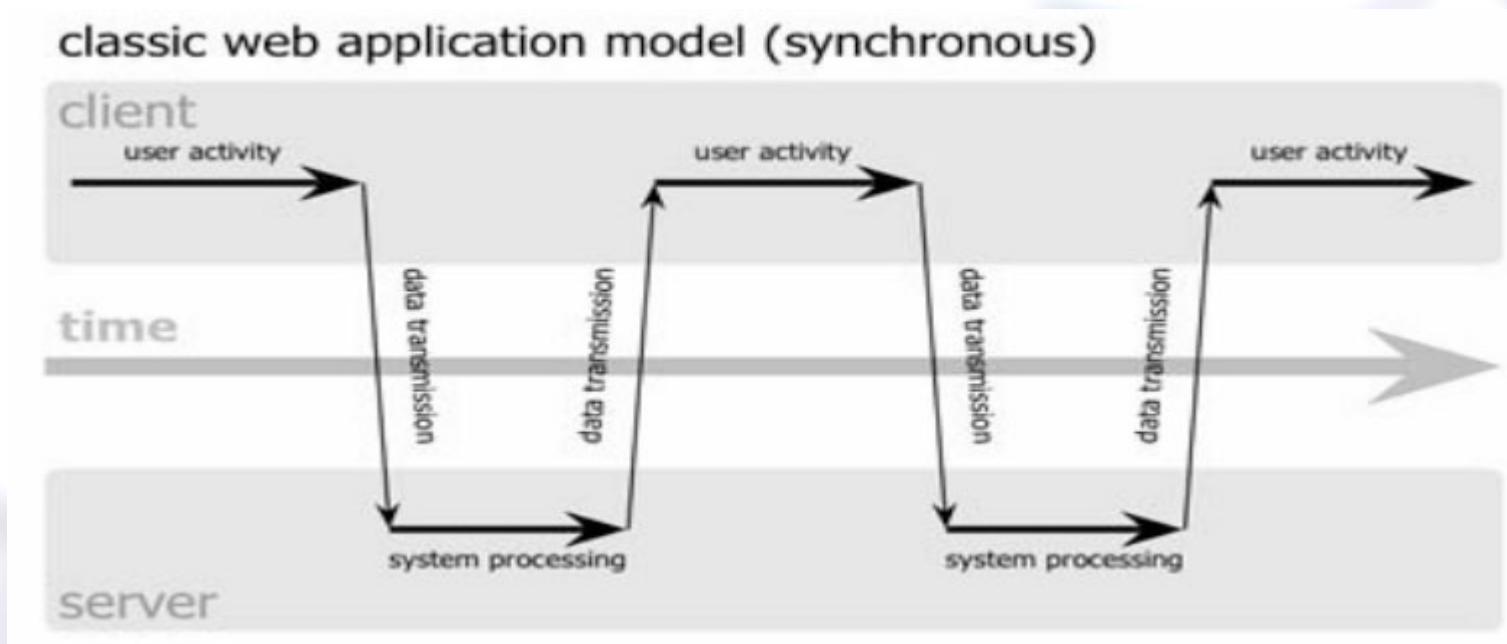
Web services concept

- XML is used to tag data;
- SOAP is used to transfer data;
- WSDL is used for describing the services available;
- and UDDI is used for listing what services are available.



Synchronous Web application model

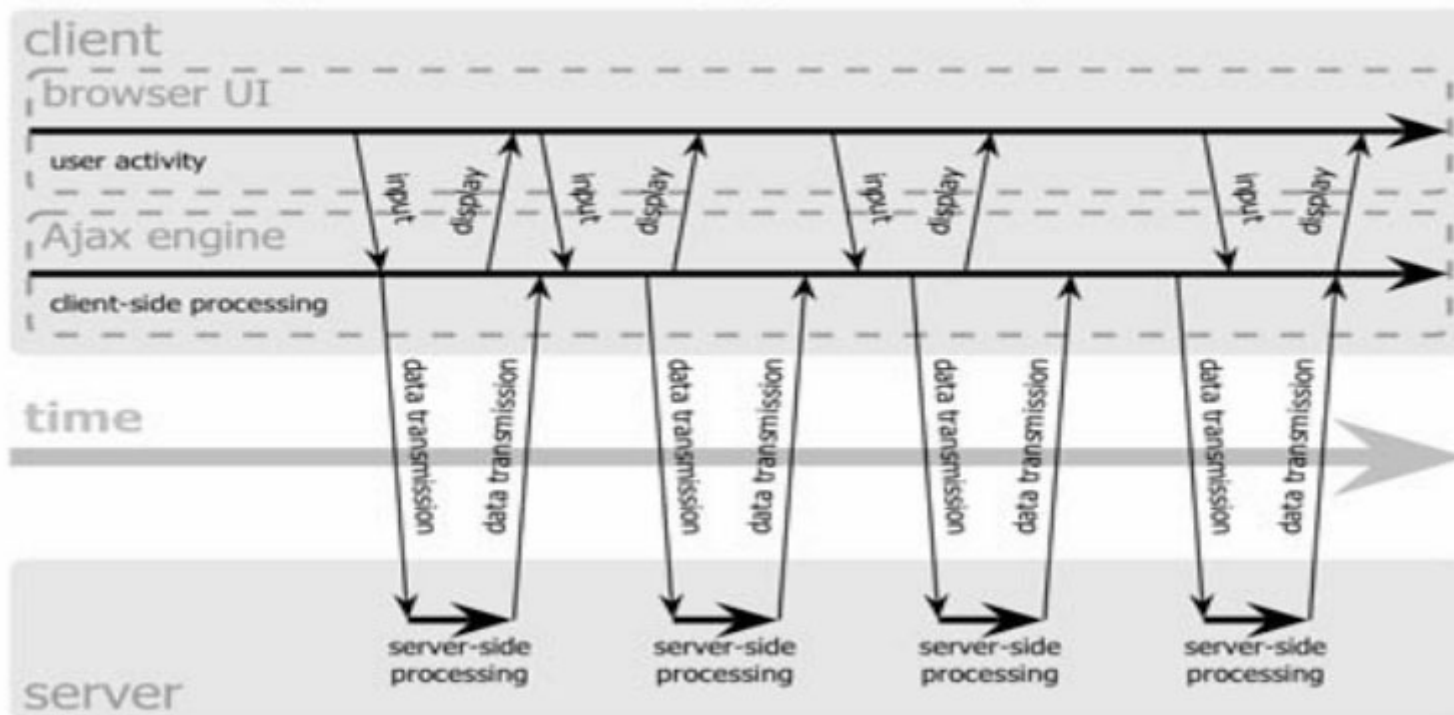
- In response to each request, server responds by sending back a new web page :



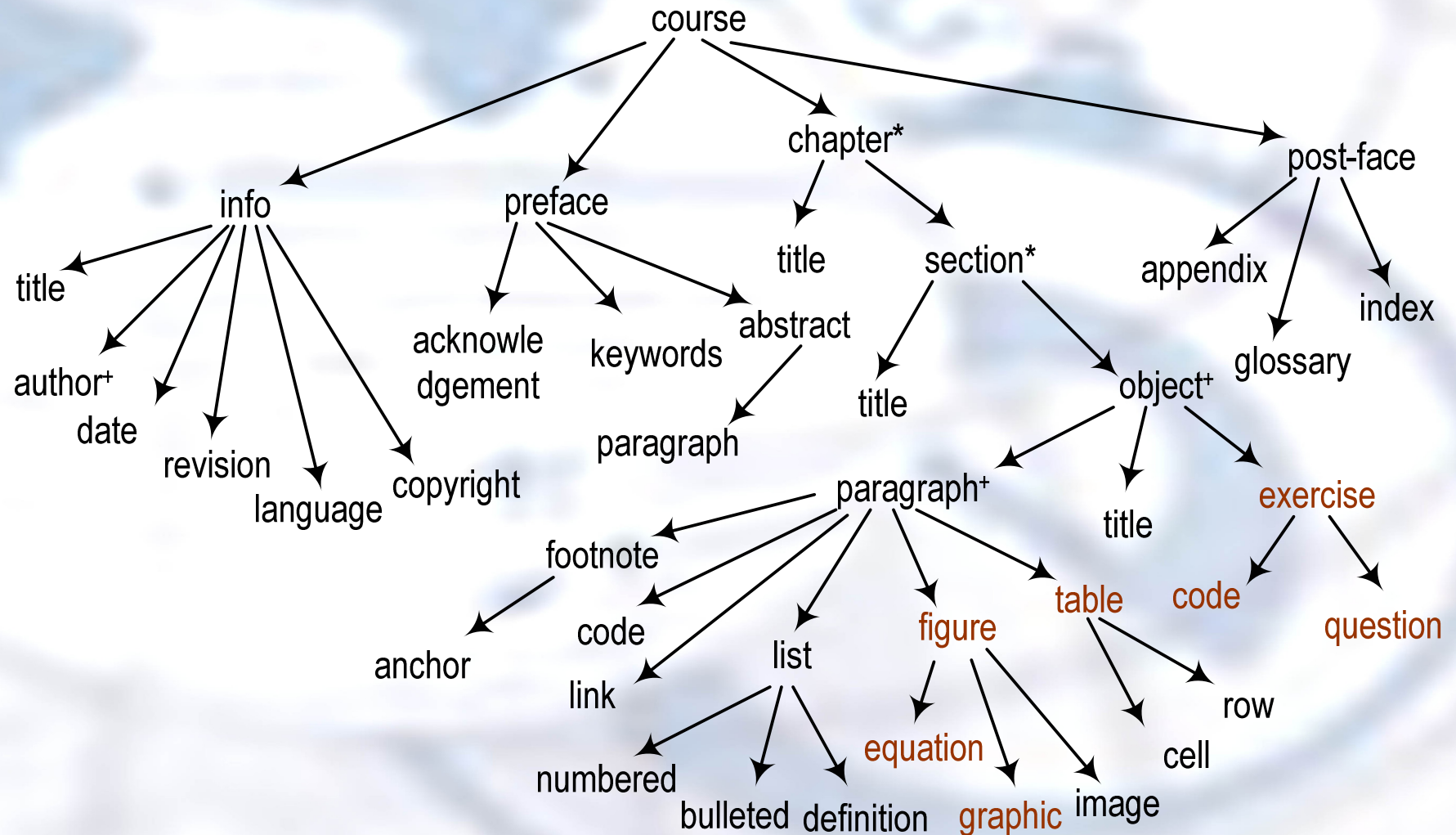
Asynchronous JavaScript and XML

- The Web server-engine retrieve and transfer only the data that is needed :

Ajax web application model (asynchronous)



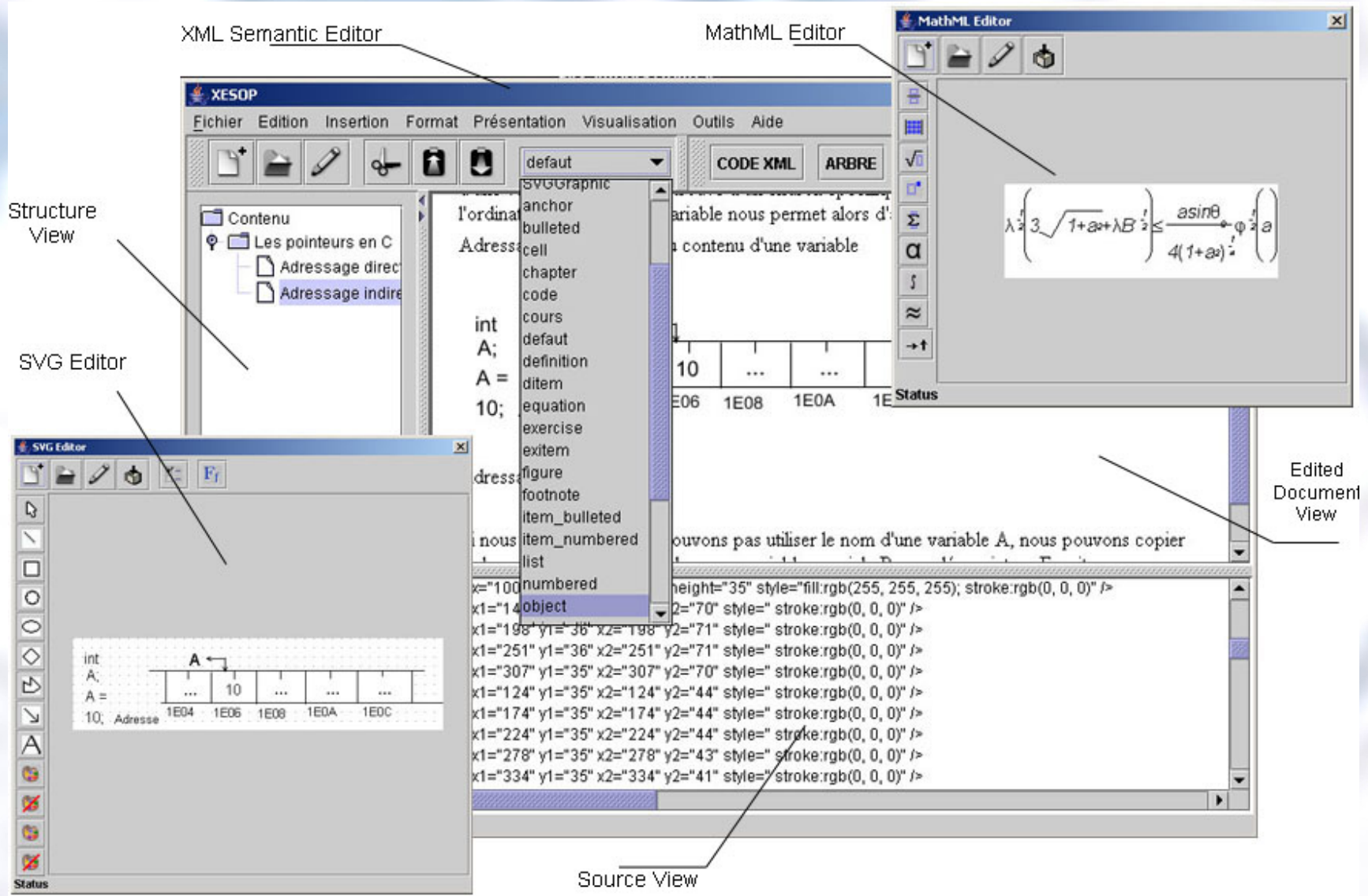
Course semantic structure



Course XML Schema Grammar

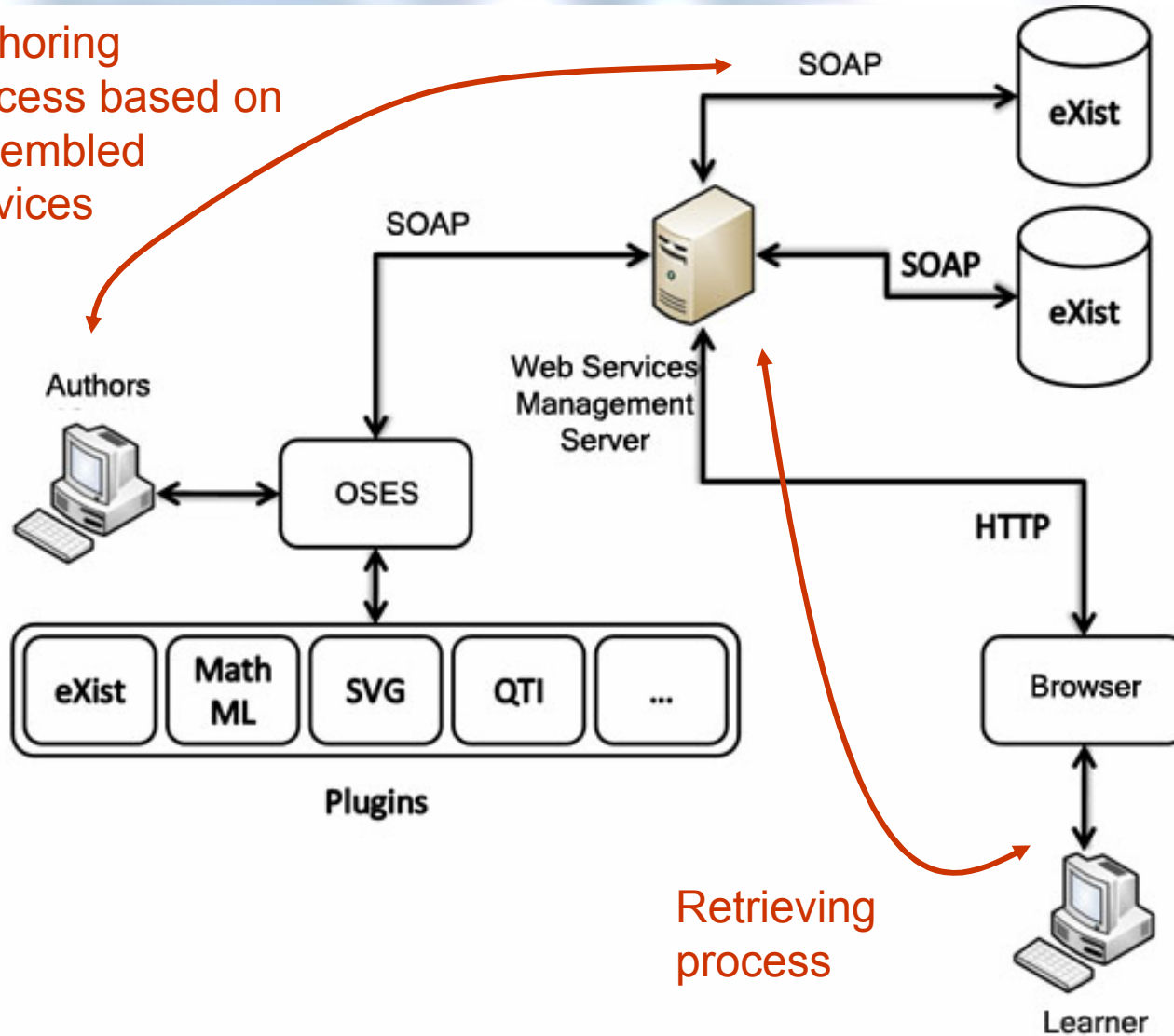
- `<?xml version="1.0" encoding="UTF-8"?>`
- `<xs:schema elementFormDefault="qualified" xmlns:math="mathml" xmlns:svg="svg">`
- `<xs:element name="SVGGraphic" type="svg:SVGGraphType">`
- `<xs:annotation>`
- `<xs:documentation>SVG graphic</xs:documentation>`
- `</xs:annotation>`
- `</xs:element>`
- `<xs:element name="inlineequation">`
- `<xs:complexType>`
- `<xs:choice>`
- `<xs:element ref="mediaobject" />`
- `<xs:element ref="math:math" />`
- `</xs:choice>`
- `</xs:complexType>`
- `</xs:element>`
- `<xs:element name="para">`
- `<xs:complexType mixed="true">`
- `<xs:choice maxOccurs="unbounded" minOccurs="0">`
- `<xs:group ref="paraclassgroupe" />`
- `</xs:choice>`
- `<xs:attribute name="type" type="xs:string" use="optional" />`
- `</xs:complexType>`
- `</xs:element>`
- `</xs:schema>`

Open XML Semantic Editor Suite



Learning Object Web services

Authoring process based on assembled services



Use case scenario (Author side)

- Provide LOs as Web services :
 - The learning content is stored in a NXDB,
 - The content is extracted and enriched by a presentation specification,
 - A SOAP message is constructed and sent back to the authoring OSES suite,
 - A Web service is in charge of publishing parsed data in the LCMS database.
 - Service-oriented XML mapping to LCMS learning content database.

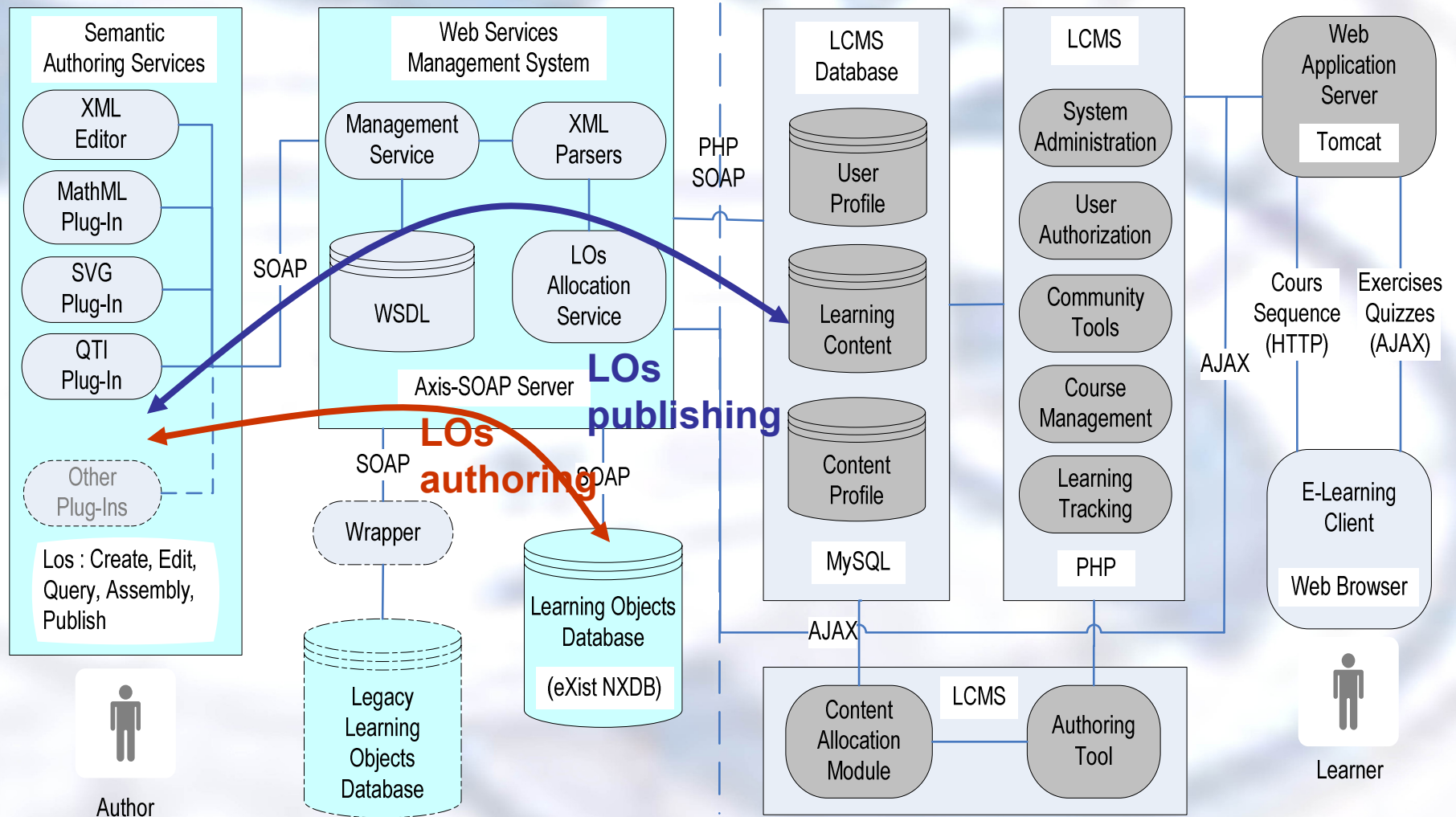
Use case scenario (Author side)

- Author's interact with *OSSES services*.
 - Store created *LOs* in the *NXDB*
 - publish a course in a suitable format (XML, XHTML, and PDF) using *WSMS services*, in the *LCMS Learning Content Database*.
 - *WSMS* can reach pedagogical content, via suitable wrappers, in an external *Legacy Learning Objects Database*.

Use case scenario (Author side)

- Web service-based data integration process
 - a mediated schema is specified in WSMS.
 - data source description is supplied to a service.
 - The data integration process is made within *OSES services*.
 - mapping between the corresponding elements of the source schema and the mediated schema

Service-Oriented System Architecture



Used Tools and Technologies

- Jaxe as XML semantic editor
- Tomcat as Servlet Application Server
- AXIS as Web services description registry
- eXist as Native XML Database
- AJAX technique as Web-service based asynchronous communication with e-Learning client
- Dokeos as open LCMS, based on PHP and MySQL

Conclusion

- This work is part of a research project supported by the French AUF (*Agence Universitaire de la Francophonie*)
- In adopting free and open source components, we believe that we provide a flexible method based on SOA that allows a high level of abstraction at the design level, and many choices at the implementation one.

Questions

- ivan.madjarov@univmed.fr
- omar.bouselma@isis.org