IMUTA - a framework for integration

Sylvester Drozdik
CIM Research Laboratory
Computer and Automation Research Institute, Hungarian Academy of Sciences
Kende u. 13-17, H-1111 Budapest
Hungary
drozdik@sztaki.hu

Abstract – **In order that enterprises could be able to exploit multimedia value at certain levels of its business software solutions must support the integration of the existing resources with multimedia extensions. Our approach of multimedia and enterprise resource integration led to the development of a framework called IMUTA where we introduced semantic web solutions as "glue" of integration.**

I. INTRODUCTION

In the vision of IMUTA we introduced an extended view of resources: for a given resource there are *related* resources and we should be able to show and manage these relationships and resources altogether.
A common example from the (manufacturing) enterprise practice would be a machining center working on a complex part. There are CAD drawings of the part, machining CNC programs, actual machining process data, scheduling, quality assurance documents related to the machining centre and the process itself. In addition we provide real-time video streams from different positions, simulation models of the machining, videoconferencing images with the shop floor workers, etc. All these things are related some ways, and in certain circumstances we have to see a set of these pieces together for example for maintenance reasons.
Supporting this idea IMUTA provides

- an integration architecture for different types of resources,
- a set of media-flow processing functions,
- a way to build up, share and modify custom views of the related pieces of information.

Through a portal-like, browser enabled web application (see Fig 1. below).

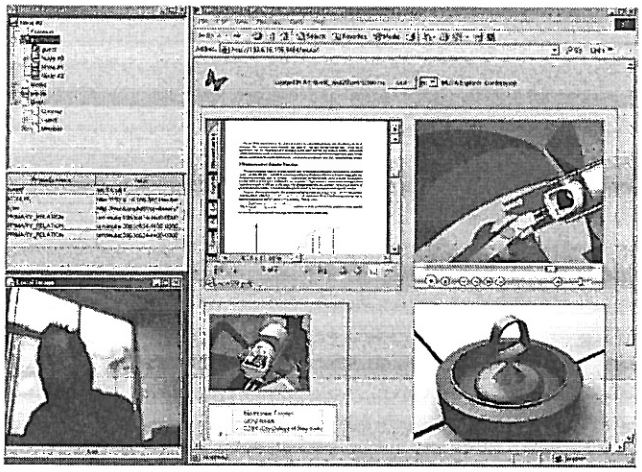

Fig. 1. Screenshot of an IMUTA client view

II. VARIOUS RESOURCES

In our approach resource could be anything accessible from the framework for example

- Different kind of documents
- Multimedia devices
- Users and systems interacting to IMUTA
- Enterprise IS components

Of course, resources don't become accessible automatically: there need some integration development. In IMUTA we applied the "*domain*" concept to refer a more or less homogenous set of resources and developed "*domain adapters*" to integrate these resources with our system.

III. MULTIMEDIA EXTENSION OF RESOURCES

Multimedia provides rich views of content. Beyond the classical textual and audiovisual media, VR models, data visualization images and increasing number of media types forms "multimedia".

We consider multimedia much more human centric as pure media solutions. In a real multimedia presentation we can show the same thing from different aspects. Additionally, most people are visual, they not just prefer images, they *think* visual.

Consequently, extending the simple views of resources with multimedia capabilities, we provide a more comprehensive, more sophisticated and more understandable views of resources.

IV. A SEMANTIC WEB SOLUTION

Without a deep introduction to Semantic Web [1] here we just quote a famous definition from the original source [2]:

"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation".

This definition is very important for us, since it promises tools to define and understand the meaning of data, which is a key issue in resource integration.

To integrate a random type of resource, the system must identify its features:

- What kind of operations the resource accepts,
- How this resource relates to others,

- What are the most appropriate tools to manipulate the resource?

Based on this features the system is able to set up the processing environment and provide an operational view of the resource.

We found RDF (Resource Description Framework) [3] and OWL (Ontology Web Language) [4] specifications as the more valuable, standardized and supported ones to represent resource types and features. So the common resource (entity) model is ontology-centric and we use the OWL language to

- formalize a domain by defining classes and properties of those classes,
- define individuals and assert properties about them, and
- reason about these classes and individuals to the degree permitted by the formal semantics of the language.

We developed ontology for each domain in the system (see Fig. 2.). In case of a new domain to involve, there we need the ontology defining the semantics of the domain and a set descriptions to describe resource specific features.

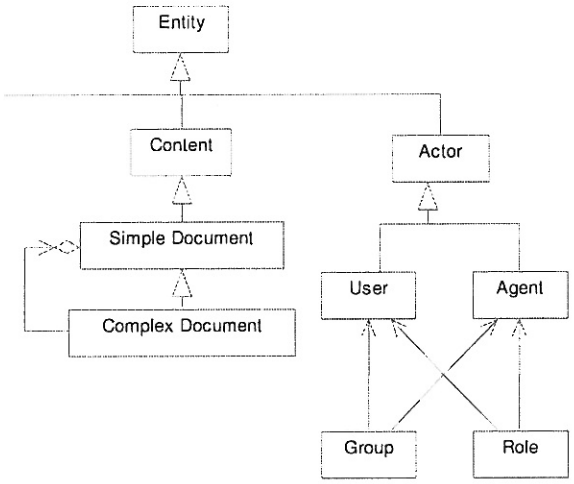

Fig. 2. A piece from the ontology used in IMUTA

Applying the semantic web concepts we adopted a solution for the representation of *knowledge* about resources. This knowledge is extendible and distributed, that is

- Our ontology and descriptions could be referred from anywhere over the internet (according to the security policy).
- We also may refer to other knowledge bases similar way.

It makes the semantic web really powerful and appropriate in our solution.

V. SOFTWARE PLATFORM

IMUTA lies upon the J2EE [5] platform, more exactly its JBoss [6] implementation, where (in addition) advanced JMX [7] features support the development (see fig 2.).

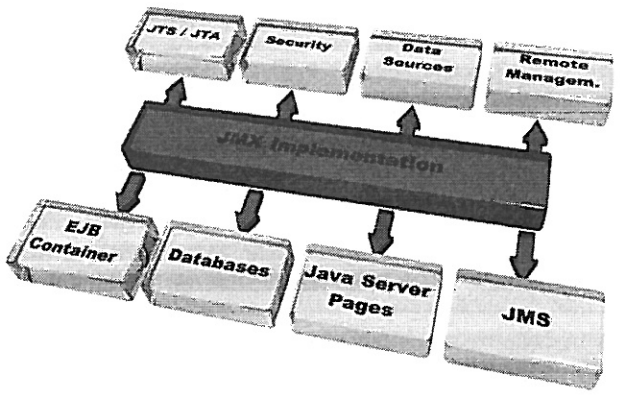

Fig. 3. The J2EE platform chosen for IMUTA

In our case the J2EE platform were a key issue and defined a way to follow. J2EE provides sophisticated standards for enterprise application integration and we can exploit these features to integrate resources into the framework:

- Database level: JDBC drivers for the common data sources.
- Message Queues to integrate with message based systems.
- To integrate with other J2EE solutions the EJBs remote interface provided.

To integrate with the less limitation, we applied Web Services [8] not just for the EAI problems [9], but as a layer upon the business logic. The Web Service (in fact SOAP) interface made it easy to integrate with Microsoft.NET applications as it was required in some cases.

We emphasis, that our development is based on pure Open Source Java solutions: there are no licensing problems and still got innovative and mature solutions. We also focused on building our framework along open standards like the W3C ones.

VI. IMUTA ARCHITECTURE

The architecture of IMUTA is built around an entity model and a semantic engine (see Fig. 4), these two elements provides the basic functionality for all the other ones.

- *Semantic Engine*: manages the ontologies, inferencing rules, resolves references to entities (resources) and responsible for identifying which one (or more) classes an entity belongs to.
- *Common Entity Model*: A repository of entities, lookup services, management of common properties and operations.

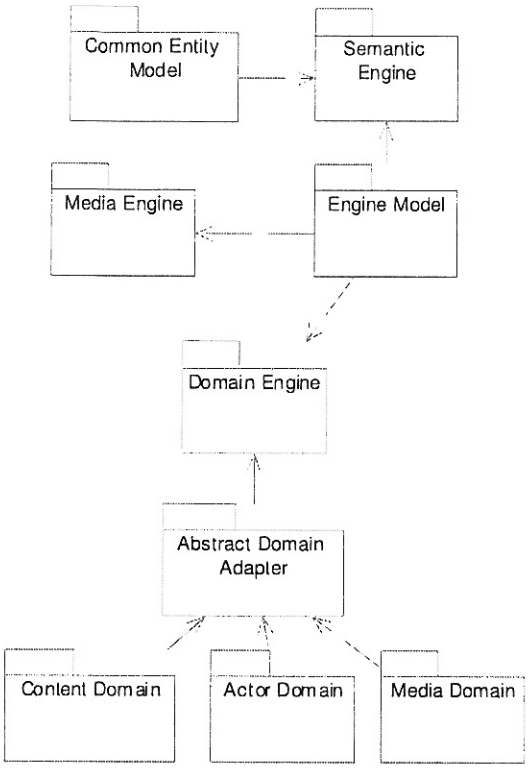

Fig. 4. Package hierarchy in IMUTA

Domains are accessed through domain adapters. There are three basic domains in IMUTA:

- *Content domain*: provides content management functionality and integrates local- and network file systems. Using DAV (Document Authoring and Versioning) [10] services over HTTP, we provided a very flexible integration model for content management.
- *Actor domain*: Actors are active players interacting from outside of the system. Actors could be human users as well as software agents. We planned the actor domain to be the base of the security and authentication functionality by defining roles and privileges.
- *Media domain*: IMUTA has a complete independent yet integrated Media Processing subsystem. This system manages dynamic, distributed media flows accessible from IMUTA too.

To integrate a domain, we defined an abstract domain adapter to derive adapter for the concrete one.

VII. CONCLUSION

The main conclusions of this development project were the followings:

- The support of enterprise resources with multimedia solutions provides much more comprehensive views and better manageability.
- Using semantic web technologies we reached quite high level of integration of resources.
- The J2EE platform and open source components form a very solid and innovative foundation of software solutions to build upon.

VIII. ACKNOWLEDGEMENT

The research project was partly supported by the National Committee for Technological Development (OMFB) under the NKFP Digital Factory contract.

IX. REFERENCES

[1] "Semantic Web" *http://www.w3.org/2001/sw/*

[2] Tim Berners-Lee, James Hendler, Ora Lassila, "The Semantic Web" *Scientific American,* May 2001

[3] "Resource Description Framework" *http://www.w3.org/RDF/*

[4] "OWL Web Ontology Language Overview" *http://www.w3.org/TR/owl-features/*

[5] "Java 2 Platform, Enterprise Edition" *http://java.sun.com/j2ee/*

[6] "JBoss :: Professional Open Source" *http://www.jboss.org*

[7] "Java Management Extensions" *http://java.sun.com/products/JavaManagement/*

[8] "Web Services" *http://www.w3.org/2002/ws/*

[9] Abraham Kang "Enterprise application integration using J2EE" *JavaWorld,* August 2002 *http://www.javaworld.com/javaworld/jw-08-2002/jw-0809-eai_p.html*

[10] "HTTP Extensions for Distributed Authoring -- WEBDAV" *http://asg.web.cmu.edu/rfc/rfc2518.html*