WebLab: An integration infrastructure
to ease the development of multimedia processing applications

Patrick GIROUX, Stephan BRUNESSAUX, Sylvie BRUNESSAUX, Jérémie DOUCY, Gérard DUPONT, Bruno GRILHERES, Yann MOMBRUN, Arnaud SAVAL

Information Processing Competence Center (IPCC)
EADS Defence and Security Systems
Parc d’Affaire des Portes
27106 Val de Reuil
http://weblab-project.org
ipcc@weblab-project.org
{patrick.giroux, stephan.brunessaux, sylvie.brunessaux, jeremie.doucy}@eads.com
{gerard.dupont, bruno.grilheres, yann.mombrun, arnaud.saval}@eads.com

Abstract:
In this paper, we introduce the EADS’ WebLab platform (http://weblab-project.org) that aims at providing an integration infrastructure for multimedia information processing components. In the following, we explain the motivations that have led to the realisation of this project within EADS and the requirements that have led our choices. After a quick review of existing information processing platforms, we present the chosen service oriented architecture, and the three layers of the WebLab project (infrastructure, services and applications). Then, we detail the chosen exchange model and normalised services interfaces that enable semantic interoperability between information processing components. We present the technical choices made to guarantee technical interoperability between the components by the use of an Enterprise Service Bus (ESB). Moreover, we present the orchestration and portal mechanisms that we have added to the WebLab to enable architects to quickly build multimedia processing applications. In the following, we illustrate the integration process by describing three applications that have been developed on top of this architecture on three R&D projects (Vitalas, WebContent and eWok-Hub). Finally, we propose some perspectives such as the realisation of an information processing services directory, or a toolkit following MDA (Model Driven Architecture) approach to ease the integration process.

Keywords:
Integration infrastructure, Service Oriented Architecture, Semantics, Multimedia Information Processing Platform.

1. INTRODUCTION

Internet is a huge, ever-evolving source of information: from 100 000 sites in 1998 to 100 millions at the end of 2006 representing 11 billions of visible web pages and a lot more of other pages coming from the so-called “invisible Web”. Recent studies have shown that over the past two years 90% of new information were digital and that it is expected in the next coming two years more new content than in the whole history of humankind. Progresses made in the multimedia domain over the past 10 years have allowed our society to fully enter the era of all digital (documentation, television, photography, telephony, etc.). With the adoption of ADSL and other large broadband technologies, several terabytes of data are now exchanged every day in the cyberspace, these data either being structured information coming from databases or non structured data such as Web pages, emails or multimedia files.

The use of new tools has now become mandatory to be able to take the best of all this information now digitally available. The analysis of large volume necessitates the automation of all or a part of the analysis, interpretation and content understanding processes. Advanced information processing techniques appear as the adequate solution for the extraction, digitalisation, normalisation, transcription, retrieval, display, representation, translation, sorting, classification, and merging of these multiple and various information. One generally admits that 80% of the available information is not
structured which means that the organisation of the information does not follow any specific rule and thus requires to be retrieved, isolated and identified prior to be exploited by advanced processing techniques. This is the reason why the software tools have to allow for the isolation and extraction of new and useful knowledge from documents and data streams which are now massively available on-line and off-line. More specifically, these techniques bring an appropriate answer for applications such as technical watch, economical watch, surveillance, intelligence, knowledge extraction and knowledge management, multi-media document management and so on.

In this context and taking into account the various needs, document mining is now considered as one of the most outstanding and promising technology of the 21st century.

2. PROBLEM STATEMENT

2-1 Motivation

In the various applications mentioned above, the main issues are information in its various representations (text, audio, image, video and structured data) and all the different processing that can be applied: acquisition, collection, storage, formatting, transformation, description, annotation, visualisation, display, structuring, classification, communication, sharing.

Numerous tools and components are now available for one of the mentioned applications. However, few tools are providing a fully functional coverage for a given application. For the sake of efficiency and development cost savings, a natural and pragmatic approach consists in choosing the best tools for their integration in order to satisfy a set of given requirements. This leads to assessing each component of the overall solution with respect to the various services that it may offer, then selecting the best component in each functional domain (the tool with the best level of service) and to consider it as one the components of a global solution covering all functional requirements. The interoperability of the selected components is then a must that might sometimes happen to be difficult to achieve since their development of were made separately by different vendors with different technologies. The integration of tools relying on disparate technologies has to be facilitated first by the establishment of a common structure offering communication and exchange mechanisms between the components and then by a framework facilitating the development of connection gateways.

One should also note that, given the innovative nature of document mining technologies, the functional components now available can exhibit very different levels of maturity. For a given functionality, the use of different technologies may be required, some of these being more efficient or more stable than the others. Information extraction can, for instance, be a simple calculation of occurrence frequency or a more complex computation based on linguistic techniques. An evaluation of the available software components is thus needed and the results of these evaluations must be regularly updated due to the quick progress of these technologies that may be immature today and very efficient in the next few years. Given this context, an integration platform contributes to keep a perennial solution by providing interchangeability of components and allowing replacing a component turned obsolete by a new and more adequate component.

With the ever-growing and increasing volume of exploitable information, the information systems require more and more resources and processing abilities. In an integration approach, the share of the resources required may be necessary. The ability to distribute these on a computer network is thus mandatory.

2-2. Requirements for an integration platform

To satisfy these different requirements, an integration platform should host and allow interoperability between functional components addressing the various needs of acquisition, processing, and exploitation of multi-media documents. This platform should provide:

- a software infrastructure with mechanisms for the integration of different tools as components of a global operational solution,
- an open and extensible middleware for the mediation between the integrated components, no matter their origins and the technologies they use,
- a connector model for inter-component communication and their technical interoperability,
• a reference model to normalise the exchange between components and their semantic interoperability,
• a solution to build processing chains involving a set of inter-operable tools.

This software platform must also be open, upgradeable and thus be modular.

Following standards is also a key element to reach the objective of component interoperability. As long as they are adopted by the majority of software vendors and consensual, the standards ease tools convergence.

3. PLATFORMS REVIEW

The term « platform » can be used to refer to very different ideas. Excluding the hardware platforms and the infrastructures, we can distinguish:
• The operating systems (Linux, Windows, Mac OS, etc.),
• The Web platforms that are similar to a complete infrastructure including hardware, software and, sometimes, networks,
• The software platforms that are generally called frameworks and that aim at easing the development for technical aspects or a specific business domain,
• The integration platform that can be considered as a technical foundation or software facilities. This kind of platform provides mediation mechanism to allow interactions between components in order to fall short of complex requirements.

In the following, we present more specifically some examples of the last two categories of platforms. They are the ones that can address the requirements we mentioned in chapter 2 and answer the following questions:
• Is it possible to build up specific applications or software systems with given requirements using a set of existing modules that deliver the needed functions through an API?
• What are the architecture pattern and the software mechanisms that can be used to allow modules interactions and collaboration between heterogeneous components coming from various origins? The aim of these disposals would be to obtain a unified and coherent application.

GATE (General Architecture for Text Engineering) [1] is an open-source framework for developing language engineering components. Plugins shall be created using the Java language only. The internal representation of processed documents is inspired from the TIPSTER recommendations and can only handle with text and annotations. Many academic language resources and components are available on various licensing basis.

LinguaStream [2] is a free of charge, for research purpose only, integrated experimentation environment to combine natural language processing algorithms. It is built on a component-based architecture to process any XML documents through a Java API. Visualisation methods help users to monitor annotated documents analyses and to build processing chains graphically.

GATE and LinguaStream are two examples of software platforms.

UIIMA (Unstructured Information Management Architecture) [3] is a framework developed by IBM for analysing unstructured content such as text. The internal representation is based on an extendable XML representation called XCAS. It comes with a SDK to built processing components. Its baseline enables to plug business components and to build information processing chain for any kind of media. Anyway most of available components are text oriented. It is now a part of Apache Incubator.

Commercial software for multimedia information processing, than can be considered as proprietary platform, tend to have a functional coverage as large as possible by integrating specific components developed by other software vendors. They often rely on a proprietary information model. They propose integration mechanisms to plug other software but they often rely on non standard integration mechanisms (proprietary scripting language, rigid API). The included software needs generally to comply with the proprietary model of the platform.

UIIMA and some commercial software could be considered as integration platforms.

The WebLab belongs to the integration platform category but relies on standards for its internal model and for integration mechanisms. These standards are described more thoroughly on the following.
4. THE WEBLAB PLATFORM

4.1. Origin and general description

The WebLab platform has been developed by EADS in order to ease the integration of components and the development of applications using media mining techniques in the frame of collaborative projects. Thanks to various research projects and also operational projects, the IPCC department of EADS has acquired over the past 5 years a technical expertise on the processing of unstructured data and the semantic web. EADS now positions itself as architect/integrator and offers its WebLab Core solution as a technical base for integration in different projects (ANR/RNTL projects WebContent and e-WokHub, European projects Vitalas and Citrine, or advanced studies for the French Dod, or also industrial applications part of large programmes or for other business units of the EADS group). In all these projects, EADS’ French and European partners have contributed to the evolution of the platform and collaborate to enhance the WebLab according to their specific needs.

The WebLab platform provides a set of services for media mining (text, image, audio and video) and solutions for intelligence applications (business, strategic, military, etc.). WebLab facilitates the integration and combination of software components (COTS or open source) providing functionalities such as Internet crawling, information indexing and retrieval, semantics analysis, image and video analysis, speech-to-text transcription, translation and so on.

WebLab refers to three different levels:

- WebLab Core is open source technical base (http://www.weblab-project.org) acting as an integration environment for the interoperability of software components within a service oriented architecture (SOA).
- WebLab Services form a coherent set of software services and elementary human-computer interface components that can easily be integrated with the WebLab Core to build a dedicated application. Most of WebLab Services are implemented by COTS, open source components or developed by EADS and its partners.
- WebLab Applications result from the integration of WebLab Services with a minimal programmatic effort thanks to the WebLab Core mechanisms. The WebLab Applications are built by composing more or less complex processing chains, services and Human Computer Interface components being able to be connected together thanks to graphical and WYSIWYG editing tools.

4-2. A service-oriented integration platform

The WebLab architecture is organised in several logical layers (cf. Figure 2: The layered architecture of the WebLab):

- Access: it enables the users to activate business processes and interact with the system in order to achieve the processes.
• Process: it allows designing and executing application such as a work flow using an orchestration engine. It contains and editor enabling to define service chain and tools to support deployment, configuration and management trough GUI components.
• Service Access: it is composed with binding components which allow connecting the multiple software elements.
• Business Services: it includes all external software elements which realise the business process such as crawling, analysing and eventually performance measurements components.
• Data: it represents the sources used as system input and the repositories where processed data is stored for further exploitation.
• Infrastructure: it contains the operating system and virtualisation tools.

A service bus allows the communication and distribution of messages between services.

Those layers are supported by vertical components:
• Security mechanisms to manage access rights to services and ensure confidentiality and integrity of exchanged data
• Technical administration and monitoring applications that allow controlling the processing chains execution (service availability, lack of errors, etc.)

The architecture is presented in the diagram below:

![Diagram](image-url)

**Figure 2: The layered architecture of the WebLab**

The access layer allows a user to invoke the services exposed by the platform. It includes a portal which offers a unique access to all the system resources (application, services and data), controls access and handle identification using single sign on approach, manages and control GUI elements (portlet), presents information through a unified and structured model, allows to customise the work space in accordance to user profile or application. Dedicated applications using rich client could also be designed to answer specific needs using programmatic interface to request a process. Access layer is finally completed by a graphical design tool which allows building the service chains and produce a process description using WS-BPEL (Web-Service Business Process Execution Language) standard [6].

The service bus takes in charge the core messaging functionality which enables the communication on the service layer. Two different categories could be distinguished for the services: "technical" and "business" services. Each service is implemented using one or several software components and each component could implement multiple services. Integration of an external software component could be achieved at any layer:
• Infrastructure: deploying several tools on the same hardware platform
• Data: sharing data sources and repositories
• Service: component collaborate to offer a service
• Process: orchestration of multiple services in chains
• Access: aggregation of GUI elements
4-3. The WebLab exchange reference model

A conceptual information model is used (see Figure 3) to define a common exchange format and ease chaining the processing services: a producer service encodes its result and provides them to a consumer service (which will decode the data and then process them). The orchestration will then be rationalised since it does not involve specific interfaces between each service. The unique data format will also reduce computational and development costs. The introduction of new services will not need too much adaptation.

The model defines the common grammar which from a technical point of view will be expressed through XML schema. It describes the structure and parts of content of any data exchanged through the bus such the multiple document types processed in the platform. In order to ease the communication during its collaborative design, the model is formalised with UML because this visual technique is more readable and understandable than XML. The complex XML types are modelled as object classes that will be used in the services WSDL definitions as parameters of operations. When it has been approved by the community, it has to be translated in XML Schema for its technical implementation (it can be done automatically, for example, from an XMI file exported by the modeller).

The semantic Web standards (RDF [4], RDFS/OWL) have been used in order to ensure the sustainability of the model, its compatibility with existing software components involved in the application domain of the WebLab platform, the capability to exploit existing domain ontologies or build with semantic information extraction tools.

A WebLab Resource, generally simply called a Resource, will be defined as any object which has a link with the final user interests or services needs, could be used by a service on the platform and could be identified by an URI.

The resource concept includes all type of entities which should be processed by several services and also any kind of object that could be useful for a service in a specific task: document, segment of document, services configuration files or ontologies, queries, etc.

Without any wide-spread standards for describing the structure of multimedia documents, we proposed a model to segment multimedia documents. It enables to add annotations pointing to an identified part of these documents (for instance, paragraphs, sentences, words for a textual document).

Figure 3: The WebLab Exchange Model v1.0
Business knowledge such as ontologies, services and users, content... are also a kind of Resource. Any kind of Resources could be annotated with Annotation using the RDF model defined by the W3C. Since Annotation is a resource, it can also contain annotations. The contained statements can be seen as meta-annotations describing, for example, how and when the first level of annotations has been created (which service with what resource or configuration).

The content of Media Unit is referenced but not necessarily exchanged through the bus to avoid congestion. Ad-hoc dedicated communication protocol is used for large content. However the common XML format allows exchanging text content or byte array which represents a small part of content. Segment will allow localising information in content.

4-4. Interface normalisation

The WebLab Core proposes also to specify the information processing service interfaces. These specifications aim at normalising the data exchange and easing the components integration. These interfaces are described in UML then transformed into WSDL.

The XML-Scheme of the generic interfaces can be included in all services. And thus, the Web Service technology enables to generate the Java or C++ API that will enable to handle the exchange model objects in the chosen technical languages.

![Figure 4: Implementation of UML model through WSDL and XSD](image)

The UML model enables to describe abstract and generic interfaces for each functionality of an information processing chain. Then, these interfaces can be instantiated to build any service we want to implement.

Following this approach, nine generic interfaces have been identified. The instantiation of these interfaces is used for defining all services. These interfaces are defined in WSDL file that can be included in all WSDL of each service of the platform.

![Figure 5: Normalisation of service interfaces](image)
4-5. Orchestration mechanisms

The WebLab platform provides the means to define collaborations of WebLab atomic services in order to accomplish a particular business objective through a complex process. To execute a business process, the platform uses an orchestration engine based on the W3C standard: WS-BPEL. This standard describes interactions between multiple services to provide a higher value services. Adding an orchestration level to the platform enables fully loosely coupled elementary services.

WS-BPEL enables the description of a complex business process by using standard operations such as service invocations (using the bus routing system), conditional blocks, loops, variable affectations, etc. The orchestration engine can chain WebLab Services and allow their interactions according to the WS-BPEL program. The resulting process uses each service on demand without worrying about their real implementation. It means that a WS-BPEL chain is exposed and requested outside the bus exactly in the same way as another service. Moreover, a process invokes a service or a sub chain using the same mechanism (no differences between classic unitary service and process invocation). Simple chains will exploit unitary services and can be exposed as simple WebLab Service using generic interfaces. Complex processes, like an indexing chain, can use chains made of sub chains and/or unitary services.

With multi-level service composition and bus capability to select dynamically services at runtime, the platform is flexible and agile. It is possible to develop a processing chain based on existing WebLab Services by specifying it with a graphical WS-BPEL editor and generating a runnable program. WebLab common model and generic interfaces ease the chain creation by enabling "pipeline" processes where a service response can be directly used as next service request.

Adding, updating or removing a service implementation has no functional impact on the whole business process. The platform just needs to update its composite process using WS-BPEL capability to assign dynamically the service reference before invocation.

4-6. Portal

The WebLab platform also integrates a Web portal providing the user with a unique access to a panel of resources and services available in a customisable working environment. In the case of WebLab, the portal relies upon the portlet technology.

A portlet is a unitary Web component producing a part of HTML or XML content that can be displayed in a portal. The set of parts of content created by the portlets of a portal is aggregated to shape a unique Web page. The figure above shows an example of a Graphical User Interface (GUI) of portal type.

Setting up such a portal relies on programming portlets that will take place in the user interface inside the portal. Each of these portlets is programmed in Java following the recommendations of JSR 168 and JSR 286 [7] which guarantees the portability of the portlets in every portal following these standards. This allows, in the case of our solution, planning a possible change of portal without having to reconsider existing GUI.
The “Service” layer of the WebLab platform includes a library of portlets that one can use to build a human-computer interface by simple assembly of existing portlets. This approach of GUI development by composition is comparable to the approach used to aggregate elementary services in processing chains. The portlets are seen as services for command and/or display; they allow a controlled access to “business” services, triggering their use and display of their results. Building a WebLab Application then corresponds to defining an orchestration of “business” services and portlets.

5. SOME IMPLEMENTATION EXAMPLES
WebLab Core and WebLab Services have allowed the development of different applications.

In the Defence and Security domain, EADS has developed an application upon this platform to provide an OSINT (Open Source INTelligence) solution. Open sources refer to information legally available without the need for special security clearance. This encompasses Internet, radio, television but also the press, CD-ROM and so on, all information that is freely or commercially available to “anyone”. The solution is useful to collect, analyse and share the information coming from this kind of sources.

In the civilian domain, WebLab Core has been the core integration platform for various research projects supported by the French government (ANR/RNTL, OSEO ANVAR initiatives) or by the European commission.

The WebContent project (http://www.webcontent.fr) aims at creating a Semantic Web platform. This platform relies on the WebLab Core. The text mining tools that have been provided by different research laboratories or small to medium enterprises are implemented as WebLab Services. Four different applications are under development: economic watch in aeronautics, strategic intelligence, microbiological and chemical food risk, and watch on seismic events.

The eWok-Hub project (http://www-sop.inria.fr/edelweiss/projects/ewok/) - Environmental Web Ontology Knowledge Hub - aims at building a set of communicating portals (the e-WOK Hubs), offering both: (a) web applications accessible to end-users through online interfaces, and (b) web services accessible to applications through programmatic interfaces. One application resulting from the integration of different components provided as WebLab Services on top of the WebLab Core is being developed focusing on enabling knowledge management of several projects on CO$_2$ capture and storage.

At the European level, the Vitalas project (http://vitalas.ercim.org/) – Video & Image indexing and reTrievAI at The LArge Scale) addresses the issue of providing professional end-users dealing with large multimedia archives (e.g. INA, Belga, IRT) advanced solutions for indexing, searching and accessing non previously (or partly) annotated content. All Vitalas multimedia information processing services are implemented as WebLab Services on top of the WebLab Core.

All these projects involving more than 30 partners have allowed WebLab Core to gain in maturity. As a result, developers of new WebLab Applications gain time during the integration phase thanks to all these WebLab facilities.

6. PERSPECTIVES AND CONCLUSION
We have presented a description of EADS WebLab platform that has led to the realisation of four prototypes in very different domains. Furthermore, the infrastructure layer, the WebLab Core, is available for download in open source at http://weblab-project.org.

The WebLab platform appears as a coherent set of technical functions that allows for the deployment, the interaction and the orchestration of media mining services that can be accessed via a collaborative portal. Its definition and use has allowed validating a set of technical choices and standards that bring satisfactory decisions to most integration problems.

Several improvements are listed on our technical roadmap:

- Enrich the definition of the exchange model and complete the specification of the generic interfaces in order to address minor points which have been brought back by some project partners,
- Expand the catalogue of services by integrating new components in order to have full functional coverage and, also, allowing the choice among several of the best available implementation according to the application requirements,
o Expand the catalogue of portlets in order to provide the final end-users with a set of GUI components allowing to access available services and orchestrations,

o Develop and set up new WebLab Core mechanisms to take into account particular technical aspects such as availability of new connectors (e.g. UIMA or CORBA), security and quality of services (access to services, information exchanged on the bus, etc.), asynchronous execution of services, management of transactions, etc.

o Define ontologies for service characterisation, for portlets and business processes by integrating together properties related to functionalities, to technical implementation, to operational usage, to experience feedback etc and use these ontologies to provide catalogues of services and portlets as WebLab knowledge base.

o Allow the use of the catalogues of services and the characterisation of business processes to envisage an optimisation of orchestrations by dynamically selecting the provider best suited to provide the expected service.

o Propose technical means to developers and integrators for building WebLab Applications more simply and more quickly (WebLab C++ environment, mapping of models of proprietary data on the WebLab model, visual building of “proxy” portlets, visual modelisation of processing chains, etc.). The improvements of the platform with respect to this last item could lead to create a new level in the schema presented onto Figure 1: The different levels of the WebLab platform. This would group together automated development tools. Following the methodological approach now used to build WebLab Applications from UML models (services, information exchanged, processes, etc.); MDA tools [5] could be integrated to allow an automatic transformation of business models (PIM) into WebLab models (PSM).

References:


