Dear COMMIUS Community member,

You are receiving this newsletter because you subscribed to the COMMIUS Community, an active interest group created with the aim of supporting and driving COMMIUS research and activities, composed by subjects for different reasons interested in the project, such as persons in charge of SMEs or involved in SME associations, ICT companies who could be interested in exploiting commercially COMMIUS results or researchers and experts in interoperability and collaboration issues, mainly interested in the project scientific results to support their research activities.

The goal of this newsletter is to keep you informed with the latest project advancements and results, and to elicit your collaboration in order to constantly improve our output and provide us with your useful feedback on our results.

If you wish to contact us, please feel free to write to the COMMIUS community e-mail address community@commius.eu.

This issue’s contents:

- Project News
- The COMMIUS System Layer
- COMMIUS as Open Source
- COMMIUS on LinkedIn
- Agenda.
What's new

Here are some of the project achievement of the last period:

- The first round of End User events has been concluded. These included a first event involving representatives of CTI-Liguria club on June 30th, 2009, a second event involving SIIT-PMI in Sept. 16th and a first demonstration of COMMIUS Software held during the project plenary meeting held in Genoa during the last October. Moreover, on March, 9th, COMMIUS was presented to SMEs in Madrid in the frame of the SalonMiEmpresa conference (http://www.saloniempresa.com/). You can check the project web site http://www.commius.eu/ for more information.

- COMMIUS has successfully undergone its second project review, held on Bruxelles on Oct 16th, 2009.

- With the conclusion of the first software components release, the project has now entered the software integration phase, and the end of which a first integrated software prototype of COMMIUS framework and Pilot Applications will be available to be submitted to end users for piloting activities in order to collect their first impressions and useful feedback for the second prototype development and final integration round.

- The first components of COMMIUS architecture are now available as Open Source on sourceforge.net (see article at page 9).

The COMMIUS System Layer

After having introduced the COMMIUS architecture at a general level in the previous issue of this newsletter, we are going to introduce in some more details the three basic layers of COMMIUS software architecture. We will start in this issue with the bottom one, the System Interoperability layer, and will continue in the following two issues with Semantic and Process Interoperability layer.

System Interoperability

System or technical interoperability includes standards, protocols and also architectures which are built on top of protocols and interoperability standards. These necessarily overlap with basic technical interoperability to enable seamless communication. In short, system interoperability refers to the ability to connect systems by defining standard protocols (e.g. SOAP, HTTP, IP) and data formats.

Nowadays, the following crucial technologies are taken into account when dealing with system interoperability:

- Service Oriented Architectures (SOA)
- Peer-to-peer (P2P)
- Mash-up technologies based on Web 2.0

Service-Oriented Architecture (SOA) is a natural evolutionary step from Object and Component based approaches. The main value of SOA approaches is that it provides a framework for matching and combining needs, as well as capabilities to address those needs.
There are three major trends in Service Oriented Computing (SOC): Web Services, Grid Services and peer-to-peer (P2P) services. Web Services build on XML standards to provide a platform for building many distributed applications. New Web Services can be created on-the-fly using any existing Web Services (software components). Grid Services came originally from the Grid Computing needs which are accessing distributed computational (grid) resources. Nowadays, this has moved slightly away from original Grid definition and includes software, data and knowledge capabilities. P2P has much success and has potential to be the most powerful trend; however, it still lacks any consensus on how applications should be built and what semantics should be supported.

Other interoperability approaches come under “Web 2.0”, which is characterised by the development of “lightweight” software (from standard technology building blocks) that can be released quickly over the Web using HTTP protocol. In addition, “mash-up” technologies use public APIs of (mostly big) firms with given infrastructures or databases to provide new services. These developments offer the means for an SME to significantly lower its “cost barrier” to entry into new businesses and markets.

Research and development in the area of SOA, P2P or Web 2.0 as well as standards in specific industry sectors brought valuable results for interoperability. However, SMEs are still excluded due technology and resource difficulties with such solutions. We believe solutions like COMMIUS, built around existing SME infrastructure such as email and web can successfully address SME’s needs and also bring or wrap-up existing solutions to the use of SMEs.

Email repositories and email activity are valuable assets in any modern, internet based business organisation. Even small companies can generate large e-mail traffic and fill e-mail repositories with high volumes of data needed to accomplish their daily tasks. Email is the second most-used internet service after Web. The following features are common in the use of emails in enterprises of all sizes, including SMEs:

- Every organisation, without exception, will have an e-mail infrastructure before it reaches the stage of developing or adopting any interoperability solution.
- E-mail communication in a modern organisation is over 78% action-oriented, according to a study. Communication is perhaps the foundation for most organisational action.
- Managers, and knowledge workers of all kinds, interact with their e-mail systems on a daily basis.

When building a solution on top of email communication, an organisation does not have to change its way of doing business when such a solution is installed and set up in organisation. Users simply receive emails as before, but additional information or knowledge relevant to the interoperability or collaborative aspects is attached, as appropriate to the email. Work to connect knowledge or context-sensitive information with emails has been done in several projects such as the kMail system: this integrates e-mail communication with organisational memories, however, it also forces users to use a special email client and lacks a closed knowledge cycle loop. Another related tool is Zimbra, which offers web based client with functionality to detect objects such as phone numbers or addresses and allows some actions on these objects. Similar to kMail, Zimbra requires a particular email client and server application and thus changes existing ICT infrastructure in organization on both client and server side. Gmail, a webmail developed by Google, supports content-sensitive advertising and some actions as “add event to calendar” with the email.

Extensive work on email processing and active context sensitive information and knowledge provision has been undertaken by Commius partner IIAS, where ACoMA and EMBET frameworks were developed.

Commius will innovate by building an interoperability service utility above existing ICT infrastructure. It exploits existing useful properties of email, such as action oriented communication, available in all enterprises and the asynchronous nature of email communication. Commius can be used to complement to existing interoperability architectures through an automatic process based on SOA, P2P and Web 2.0 mash-up approaches. Moreover, standards as ebXML can be started upon receipt of an email by invoking an appropriate interoperability module to process information and data extracted from the email. On the other hand, if no Commius is installed, when received an email will still be human readable (e.g. by using XSL template within attached XML message) and thus can be processed manually if needed. In addition we can support interoperability tasks requiring human interaction or intervention.
COMMIUS Architecture’s System Interoperability Layer

COMMIUS System Interoperability layer focuses on designing and developing a basic interoperability infrastructure over SMTP. It supplies methods and components to provide data, information and functionality to be used by interoperability modules and other interoperability layers inside COMMIUS architecture, namely the Semantic and Process interoperability layer.

The COMMIUS platform is hooked through the System Layer into the mail server or mail client, similarly as email antivirus programs are used at the server or desktop side. This way the system can be used within any email client or even mobile device, without requiring changes to working practices or the adoption of new tools. By combination of server or desktop use, users can ensure that security and privacy issues are taken into account. Users and providers can be aware of what data is shared, passed via the communication.

Email communication is passed through the COMMIUS system, processed, and additional information is added to email messages in form of links in HTML or text attachments. This additional information contains relevant information and knowledge, hints or links to business resources such as document repositories, databases or information systems needed in detected business context. Business context is detected from email using semi-automatic pattern based semantic annotation using predefined regular expressions patterns.

When checking new emails, a user will receive a modified email message, with embodied information. Please note that inline text attachments are directly displayed in most of email clients and they appear as part of an email message, however they do not change the email message itself. We also allow possibility to change email into HTML message with additional info and add original message as attachment. Thus the users can configure COMMIUS according to their needs depending on what devices they’re using, the email client with which they send and receive emails, or depending on user preferences.

The following picture illustrates the parts of COMMIUS architecture which are part of the System Interoperability Layer.
In summary, the functionality provided by this layer include:

- **Connection to Email Infrastructure.** Technical tasks, such as connecting the COMMIUS architecture with widely-used email servers such as Exchange, Postfix, Qmail or Sendmail, are addressed by the COMMIUS System Layer.

- **Message processing, text analysis and information extraction techniques.** Information analysis and extraction is based on advanced pattern matching techniques (regular expression, XPath, XML data), information retrieval methods, POS tagging, NLP methods, and statistical analysis, identifying and developing the most suitable methods, based on analysis of content involved in (interoperability) communication. Information analysis and extraction is also used by semantic annotation and other interoperability components from the Semantic and Process Interoperability layers.

- **Connection to existing SME infrastructure.** An important task consists in connecting to existing SME infrastructure(s), document repositories and legacy systems using system connectors built on standard interfaces, such as web services, SQL-based database access, XML-RPC or file access. The COMMIUS System Layer connects connectors to access and feed data to intranets and internet applications using Web 2.0 technology, such as URI based access or wrapping of HTTP requests. Connectors are used by relevant interoperability modules, and in semantic and process interoperability layers to provide required connections to enterprise information resources. Security and policy concerning information access are considered and employed where applicable.

- **Message post-processing** covers the inclusion of active information elements (URLs of resources, action buttons, etc.) relevant to fulfilling the interoperability task(s) relayed by the message.

In the following, we examine each of these in some details.

---

**Connection to Email Infrastructure**

There are two basic approaches to connect COMMIUS to the e-mail infrastructure: Using either client-side or server-side connection. In the case of client-side connection the software responsible for e-mail processing is installed on a client’s computer, while the case of server-side connection, the e-mail processing and e-mail enrichment is enacted on the server upon message receiving or sending. There are further possible categorizations. One such important categorization for us is that the server-side connection might be either hosted on a dedicated server or plugged-in into an existing e-mail server system.

In the case the EGP (Email Gateway Plugin), the COMMIUS component in charge of connection with email infrastructure, is integrated at server level, the communication between EGP and the server is enacted in SMTP (Simple-Mail Transfer Protocol). In the case of client-side version of EGP connection, the EGP is deployed at the user’s desktops. POP3 protocol is supported for the client-side connection.

The main difference between the server and client-side connection is the place where the email is enriched. Client-side connection can be viewed as distributed while server-side connection is more centralized. Consequently, client-side EGP enables e-mail processing only for a single user while the server-side connection enables e-mail message processing available for more e-mail recipients while naturally providing higher support for collaboration and knowledge/information sharing.
Information Extraction

In order to provide useful recommendation, we need to identify objects and object properties in the text so as to formalize email message content and context. To this goal, in COMMIUS we use Information Extraction and semantic annotation techniques. In our solution we use the Ontea annotation tool, developed in the course of a previous research project, and now available as an Open Source framework (http://ontea.sourceforge.net/). Ontea is based on regular expressions, but other advanced NLP (Natural Language Processing) and Information Extraction tools and approaches can be integrated easily by producing key-value pairs.

The annotation tool identifies objects, their properties or their position in the text by applying patterns (regular expressions) on a text. The input of the method is the mail text and the set of defined patterns, and the output is the set of extracted key-value pairs, which can be chained by various transformations such as lemmatization, relevance identification, system connectors’ transformation or transformation into RDF/OWL ontology individuals if needed. The power of this approach is in its simplicity compared to more advanced but heavy solutions such as GATE (General Architecture for Text Engineering, see http://gate.ac.uk/) as well as ability of transformation chaining and connection with information system environment (databases, documents, intranets, and internet).

The aim of information extraction is thus create and share patterns for object and their properties. We believe the best approach to be used for information extraction will be pattern based extraction for several reasons:

- Pattern can be adopted for enterprise business needs. For example product codes can differ for different companies.
- Patterns can be defined, improved, evaluated and shared for group of SME or community around similar business model or industry type.
- Patterns are applicable for different languages without need of advanced NLP tools.
- Email content contains many objects extractable by patterns, such as email addresses, phone numbers, people names, company names, dates, websites, addresses or other contact details. In addition, social networks, interaction, message passing can be extracted from email headers using patterns.
- Interoperability content in general usually contains many objects and properties which can be extracted via patterns such as amounts, money amounts, product codes, bank account numbers or customers.

One of the possible evolutions of this work is toward intelligent pattern creation with the user involved in the loop. Meanwhile patterns (e.g. regular expression) need to be defined manually and success of extraction depends much on how well the patterns are defined and tested. On the other hand such solution is very powerful in enterprise environment where business specific patterns need to be defined to identify business objects. Patterns can also be shared within a community when defined and tested for common types of objects such as people names, companies, geographical locations, addresses or contact details. In this case, one can use them directly and know its quality and expected success rates. When having such tools for detecting needed objects and their properties, they provide us the context for recommendation system.

Social Networks Extraction

In terms of Information Extraction, particular interest is covered by the extraction of Social Network information: E-mail communication analysis allows extraction of social networks with further connection to people, organizations, locations, topics or time: This information can be extended into a graph of relations among elements detected in the messages, segments of message parts or relations in the conversation threads.

We believe we can use such graph to identify the meaning of a message in terms of the most relevant elements or business objects related to the message itself, even if object does not have to be directly extracted from text. For graph inference we would like to use spread activation in similar way as IBM Galaxy in Nepomuk project (http://nepomuk.semanticdesktop.org/) We believe such approach for inference is especially good in the way we using COMMIUS as a kind of a recommendation system.
The use of social network within the email for better email management was already proved to be valid in Xobni System (http://www.xobni.com/), which use social network to suggest relevant contacts, attachments or email messages for the user. While Xobni can provide general suggestions based on generic social network, in COMMIUS we can provide suggestions based on general, interoperability specific and application specific context, in case COMMIUS is customized and connected with enterprise a legacy systems.

Connection to Existing SME infrastructure

An important task within the system interoperability layer is to connect the COMMIUS system to existing SME IT infrastructure. Legacy systems utilized by collaborating SMEs are final endpoints. COMMIUS should connect and provide their interoperability. Document repositories, relational or XML databases, spreadsheet documents, intranet applications, complex ERP systems are just a few examples of legacy systems used by SMEs to store their data. COMMIUS System Connectors are components providing access to those legacy systems for the rest of the COMMIUS system deployed at SME. Other COMMIUS components can utilize system connectors via unified interfaces, which provide necessary transparency hiding the complexities of interactions with legacy systems. COMMIUS exploits the information from SME infrastructure to annotate and, finally, enrich incoming e-mails with relevant data. In addition, System Connectors invoke operations on legacy systems, as required by business processes, managed by process layer of COMMIUS.

Technologies used in enterprise environment to address legacy systems and data integration, such enterprise information integration (EII) technologies or Enterprise Service Bus (ESB) technologies, are often too heavy for SMEs to operate. Lately, inspired by Web mashups approaches, data mashup frameworks became increasingly popular within research as well as enterprise. Data mashups aim at gathering data from multiple, independent sources. Data is then merged and presented to the user in a coherent form. COMMIUS System Connectors components can be viewed as a data mashup framework for SMEs.

In order to support interaction with legacy systems beyond the boundaries of COMMIUS pilot scenarios, System Connectors will be easily extensible and configurable to provide the needed flexibility. Also, two security issues have been considered for systems connectors: usage of security tokens required to access legacy systems and authentication and authorization to invoke system connectors.

Common legacy systems required in COMMIUS user stories are:

- SQL databases
- Spreadsheet Applications
- Web applications
- Document Repositories
- Legacy enterprise systems (ERPS, order management systems, etc.)

In terms of architecture, a System Connector Logic subcomponent is responsible for pre-processing of input data, retrieval of data/information from legacy systems and post-processing of extracted data to the required form, while Module Specific Configuration defines parameters required by Legacy System Client as well as System Connector configuration (e.g. RDMS System Connector configuration may contain RDMS client configuration (server, port, database) and definition of a query to be issued from the System Connector). Finally, a Legacy System Client provides mechanisms for connecting and retrieving data from legacy system.
Message post-processing

To conclude this tour around the COMMIUS System Interoperability Layer, we'll examine the Message Post Processing (MPP) component, which is responsible for post processing of email messages. This includes message parts composition as well as composition of information and results from other interoperability layers and interoperability modules. The message is modified to include relevant information and active information elements (links, buttons) needed to fulfil interoperability task related to this message.

The MPP component represents the last component in the chain passing trough system, semantic and processing layers visualizing its outputs. Such outputs can require different level of functionality and complexity. The integration of outputs from different components to a single easily understandable user interface is the main goal of the MPP component. MPP comprises of the two visualization sub-components according to the resulting visualization application:

- Email output is usually built as static HTML or text content that can be easily incorporated into any email client. The original email message can be enhanced by additional information or the email is reprocessed and the information is incorporated directly into the email content.

- Web browser output provides information content is built using mash-up technologies. The cornerstone of the visualization sub-component is Google Web Toolkit (GWT) library that provides basic application interface for building dynamic user interface (including layout, styles, application logic, etc.). GWT framework provides a novel approach of web application development according to AJAX methodology.

According to the usage scenarios of COMMIUS project, the message post-processing component should be deployable as server application as well as client application. Such different scenarios bring many pros-n-cons for particular deployment, but the main reason is to allow user(s) to control scalability and security. First application design of MPP component counts on the Jetty application server that is suitable mainly for client-side deployment, but is also usable on server-side. Later MPP should be run from appropriate application server residing in an OSGi container. The MPP component lays on the modular design, where the core of MPP application is designed and implemented as main layout and visualization component which allows loading modules from separate java archive files. This allows separating the development of general GUI design and specific module visual widgets. This kind of modularity is solved within the scope of MPP module and should not be misinterpreted that this feature is a part of OSGi framework.
COMMIUS as an Open Source framework

One of the main objectives of the COMMIUS project consists in delivering a platform for interoperability and collaboration as an Open Source framework. The COMMIUS solution, suitable for seamless integration within a network enterprise environment and focused on SMEs’ needs, will be modular and adaptable to different ICT infrastructure standards (which is often quite low in many SMEs), and it will be able to evolve with enterprise interoperability needs.

Open Source is a key issue for the COMMIUS interoperability solution and will support new business opportunities for SMEs in Enterprise Interoperability markets: From the point of view of the SMEs end users, the choice to adopt Open Source allows to reduce the software licensing costs, contributing to the project goal of almost-zero cost adoption. From the point of view of software development, COMMIUS release as Open Source will allow consortium partners as well as third parties in all Europe to exploit and extend the platform, or to supply services based on it, in order to produce business-specific applications. All this will contribute to the overall strengthening of Europe position in interoperability industry and science.

Licensing was defined in a way to allow future interoperability modules developers and providers to decide on licensing and related policies for their new interoperability modules which may be pluggable to the open source Commius interoperability infrastructure, or obtained modifying and extending the Commius modules.

The licence of choice for Commius framework will be Apache v2. These license has been chosen in that it allows both open source distribution and possibility to build commercial solutions on top of it, allowing a wide variety of downstream uses in both the closed and open source worlds, it includes a number of provisions that make it particularly useful for open source projects (trademark and patent provisions, provisions governing the submission of new code contributions, and so on...) and it turned out to be compatible with the many Open Source external libraries used to develop COMMIUS code.

In order to exploit potential collaboration of external independent developers or researcher in the field of enterprise interoperability, and to improve project knowledge dissemination among the widest audience, the partners have decided to anticipate the release of the COMMIUS framework as Open Source, which initially was foreseen to be done not before the end of the project. A project page has already been opened on SourceForge, one of the major open source code repositories, hosting more than 230,000 projects and having more than 2 million registered users (data updated to Feb. 2009).

Two components of COMMIUS software architecture have already been released, namely the Information Extraction and the Semantic Annotator. The COMMIUS SourceForge project page is accessible at the following link:

http://commius.sourceforge.net/
COMMIUS on LinkedIn

If you are member of the LinkedIn social network, feel free to join the newly created COMMIUS group, open both to project partners and community members, besides anyone interested in COMMIUS and its achievements.

If you are interested in joining in, search for “COMMIUS” in LinkedIn group directory, or directly go to the following link:

http://www.linkedin.com/groups?gid=2767749&trk=hp_side_g

Agenda

Here are some of the events that are foreseen in the next period:

- Development activities will begin the production of the first prototype of integrated COMMIUS platform and Pilot Application.
- On March 24th the first COMMIUS piloting session has been schedule in Fedit’s venue (and with the cooperation of ATOS ) in Madrid, involving a number of SMEs,
- The third project review will take place in Bruxelles on May, 5th, 2010.
COMMIUS is a Specific Targeted Research Project (STREP) part-funded by the EU under the 7th Framework Program, ICT priority, Grant Agreement No. 2138