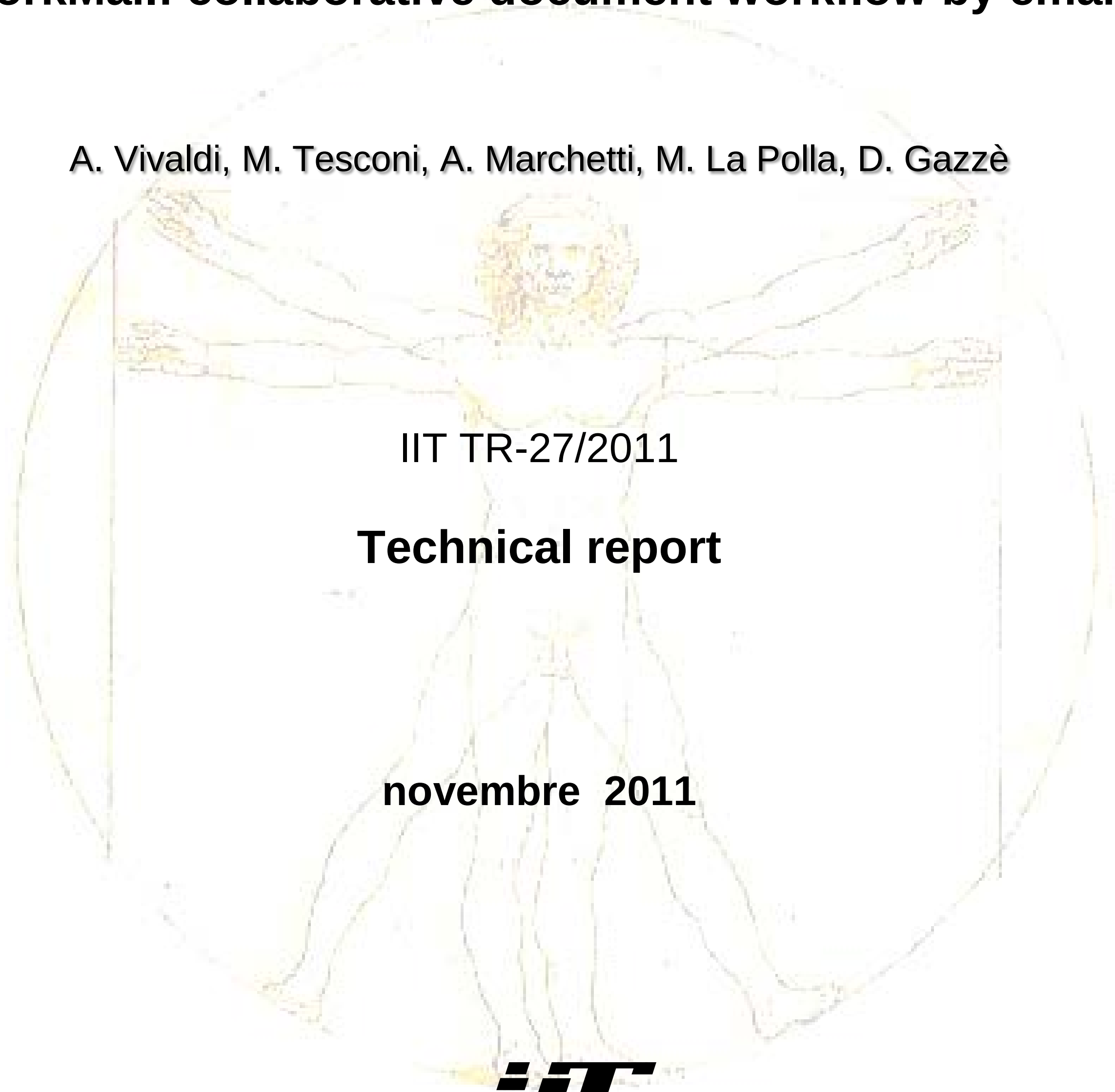
Consiglio Nazionale delle Ricerche

WorkMail: collaborative document workflow by email



A. Vivaldi, M. Tesconi, A. Marchetti, M. La Polla, D. Gazzè

IIT TR-27/2011

Technical report

novembre 2011

Istituto di Informatica e Telematica

WorkMail: collaborative document workflow management by email

Autori: Andrea Vivaldi, Maurizio Tesconi, Andrea Marchetti, Mariantonietta Noemi La Polla, Davide Gazzè.

Classificazione ACM:

H.4.1Office Automation: Workflow Management

I.7.1 Document and Text Editing: Document Management

Keywords: Documental Workflow, Drupal, PHP, Web 2.0, Web Technologies, Office Automation

Abstract

Processing documents is a critical and crucial aspect in an enterprise environment. The management of documents involves several people and many times becomes a long and wasting-time process. Many systems of document workflow have been proposed but usually they are too rigid and complex. Therefore we have developed a document workflow engine based on the email paradigm. When a user wants to make an order, a request of authorization and, in general, any kind of procedure that involve a document, starts her/his request by filling in a form and sending it by attaching it to an email. To this purpose the user has to use our web application that appears as a normal webmail client.

Our solution overcomes the actual limitation in the use of document workflow software, especially for what concern the user experience; with our system there is no need, for users, to learn the functioning of a new framework. In addition, users with different roles have different customized view of the document. According with the roles of the users, we trained the system to suggest to the user, at each step, a possible receiver of the email. Currently this feature is

based on the fact that the system knows in advance the flow associated with different type of documents.

As improvement, we will perform a statistical analysis of interactions between senders and receivers. This analysis will be used to improve the suggestion mechanism: the system will learn the most frequent interactions for each user, depending on the history of previous flow and the document involved. Exploiting these information, the suggestion mechanism will advise to the user the possible receiver of the document.

Contents

Introduction

The problem of processing documents is a critical aspect in the enterprise productivity ([1], [4], [7], [8], [9], and [10]). In fact the management of documents involves different actors with different roles, possibility of decentralized working environment, different tasks and responsibilities.

Many enterprise operations can be viewed as a series of steps involving the filling out of a form representing the document. Furthermore these operations can be concurrent. It is possible to define all the activities related to documents in enterprises environments as Document Workflow (DW).

A workflow is “the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules” [16].

With the term Document Workflow (DW) we refer to a particular workflow in which all activities are related to document’s compilation. Therefore, a DW can be viewed as the automation and administration of particular documents procedures ([2], [4], and [10]).

Through a DW, a document life-cycle is tracked and supervised continually and the document travels among agents who essentially carry out the pipeline receive process and send activity.

Due these features, a DW can resolve problems related to document management in enterprises.

State of the Art

Nowadays there are many web-based tools for the definition of document workflow. These tools are often referred as Enterprise Content Management (ECM). This term is used in the context of Content Management to refer to solutions that combine conventional tool for Content Management and web-based components that perform traditional archive, document management and workflow functionalities [14].

In the rest of this section we will briefly survey some of these tools.

We can distinguish between tools that offer ECM solutions that include document workflow facilities and tools that implement only document workflow systems.

In the group of ECM with document workflow solutions we can mention, for instance, Alfresco. Is an ECM platform built on open source Java-based software that offers both document management and workflow capabilities. The documents can be shared or modified cooperatively, exploiting shared folders (like Gmail folders).

Disadvantages come from the fact that these capabilities are too basic, in terms of supported workflow patterns and design, to serve specific and sophisticated needs.

In the same category of Alfresco, is Nuxeo Enterprise; this is the ECM product of Nuxeo suite, distributed under LGPL open source license. Features of Nuxeo are very similar to the Alfresco's ones: documents can be captured from an email or dragged and dropped from existing folders.

The workflow engine is based on jBPM (Java Business Process Management) and gives the possibility to extend integrated workflow services.

Concerning document workflow system, not ECM solutions, an example is Doqui, an Italian open source project targeted to public administration offices. Doqui offers a document workflow system that allow public administration to organize, archive and share digital documents. This is achieved with a set of different applications that implement single functionalities like sharing, archiving, etc.

Others examples are Cuteflow, an open source workflow system that allow document circulation, and DocMGR, a web-based Document Management System. While in Cuteflow, users can define "documents" which are sent step by step to every station/user in a list, DocMGR allows the storage of any file type, and supports full-text indexing of the most popular document formats.

Limits of the Document Workflow Systems

As emerge from the previous state of the art, current solutions for document workflow have some limitations. First of all, with both ECM solutions and document workflow systems, users have to learn how to use a new framework. To this purpose, a considerable amount of time can be needed. Moreover, these solutions suffer of the same problems related to generic workflow management systems.

As said in [18], the workflow system categorizes, formalize and automate work that often as a fluid and unpredictable nature.

Furthermore many empirical studies by Joostens ([19]) relates workflow management to the organizational structure type defined by Mintzberg ([20]) stressed that, in addition to the positive impacts, there are many possible negative aspects [18] .

The nature of that includes the rigid procedures, the reducing of learning by employees (because the steps are pre-programmed), the reducing of motivation of worker (works more mechanical) and, finally, the underestimation of the importance of human communication.

From this it’s possible to conclude [15] that production and administrative Workflow Management System (WfMS) are suitable for routine situations, but not in the area of knowledge work. This because in this particular area it is not possible to define a precise flow beforehand there is a need for communication and collaboration between workers. So emerge that actors choose their next steps one at time and then for this situation is very important a more flexible system (such as ad hoc workflow system and/or collaborative tools), but these systems are relatively “unspecified” ([21]).

Document workflow by mail - WorkMail

To design an efficient DW engine we start from some key aspects: (i) we need a simple tool for collaborative document elaboration; (ii) one of the most used tool in an enterprise environment is the email.

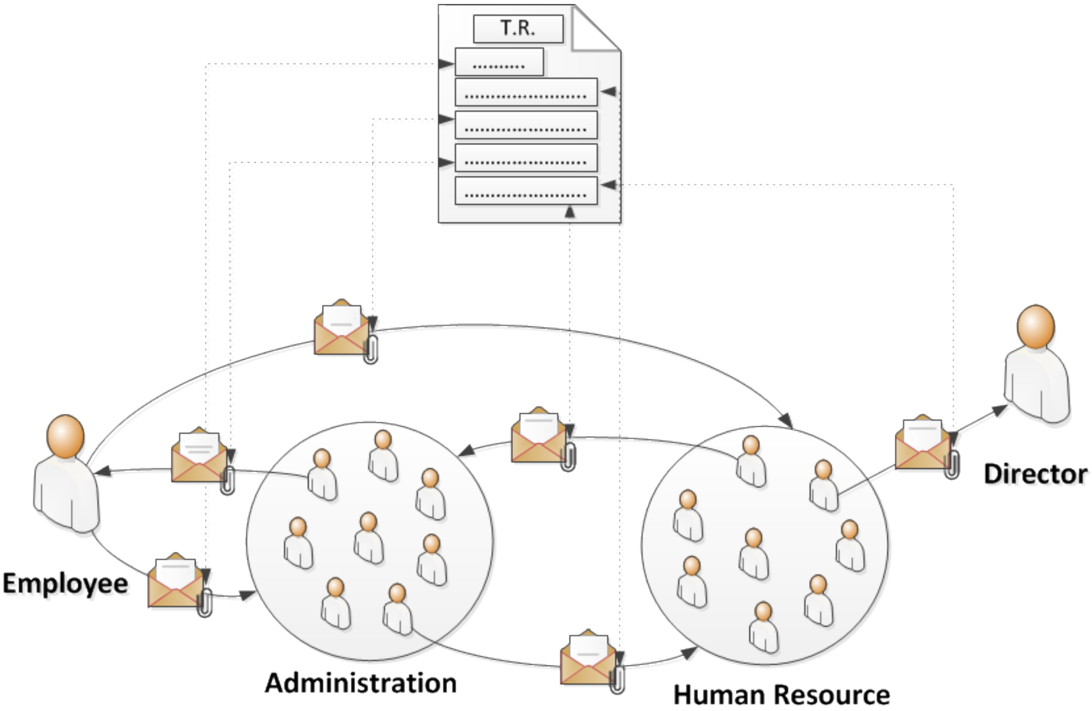


Figure 1 - Conceptual idea

The use of email is very useful in an organization environment due several reasons: users can provide quick answer, easily communicate with other people in the organization, without physically meeting them or distribute documents and information in an easy and paperless way. But the email paradigm is not sufficient to manage a DW. Despite these advantages, we have to take in account some drawbacks: they are impersonal and could be often misunderstood; for complicated question, to answer could be a time-consuming task. Moreover, the attachment cannot be modified after sending.

We use the paradigm of email as base for a new approach for a document workflow system: WorkMail. WorkMail is a web-based application that tries to combine the potential of DW with the flexibility of email; in this way we exploit the advantages of the paradigm and exceed problems related to above-mentioned drawbacks.

One of the most important advantages of our solution concerns the user experience. Due the fact that we exploit the email paradigm, users can use WorkMail easily, without learning anything about the functioning of the system. They don't need to learn how to use a new document workflow system but can continue to send email with attached document as usual; then, starting to use WorkMail is not a wasting time process.

In WorkMail users are important also for the role played in the company or in the organization. Every enterprise, organization or company have an organization chart that provides information about employees and their assignment and responsibilities. Typically, these distinctions are based on the specific office considered: employees in the administrative office have different responsibilities from human resource staff, for instance. Furthermore, in the same staff it is possible to distinguish between directors, employees, assistant, etc.
We use these differences between users to handle with the document and his fields.

The document and his modification are key aspects of WorkMail. In our solution, a document is used as attachment of an email that moves from one person to another one. Differently from the standard, in WorkMail the document is a special attachment: it is composed by different editable fields; depending on his role, different users can visualize and modify, at each step, different fields of the document.
This represents another of the most valuable advantages of using WorkMail: the attachment is unique. To modify a document, is not necessary to create a new one and attach the revised version as new attachment, but users can modify the existing one.
Figure 1 shows the base concept of WorkMail.

To illustrate the usefulness of our solution, we propose an example of usage case. We consider as case study the process of a travel request (T.R.) in a research institute, the IIT-CNR.
In our scenario an employee that have to travel for a conference or a meeting, have to start a procedure of travel request. This procedure involve :

- the employee's manager for a preliminary authorization
- the administration office for what concern the budget effort

- the director for the final authorization
- the human resource office for closing the procedure.

The flow related to this procedure is resumed in Figure 2.

In each phase of the process, users have to fill a travel request document, providing different kinds of information. The key aspect is that some of these information are known only by specific users with specific role. For instance, the information about budget are provided by the administration office because these are information that an employee doesn't know.

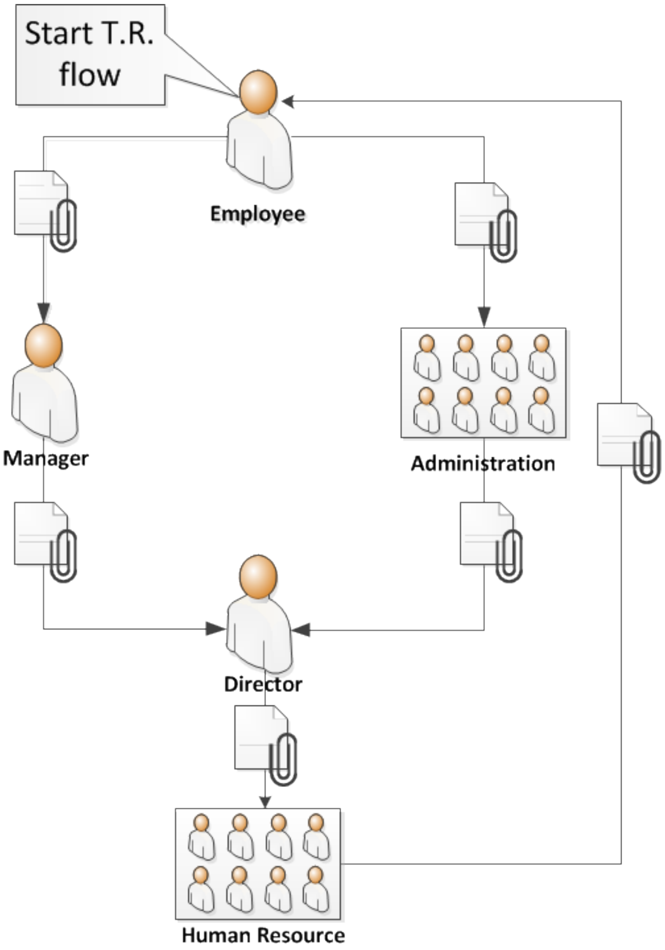


Figure 2 - Travel Request flow

This is a typical example of collaborative editing of a document. If the same procedure was performed using the standard email, at each step, the user have to

- read the current version of the document
- create a new document with the already known information
- fill the fields of the document needed at this step

- attach the new version of the document
- send the email at the next user.

Architecture and Implementation

In order to implement our solution, we have designed the architecture showed in Figure 3.

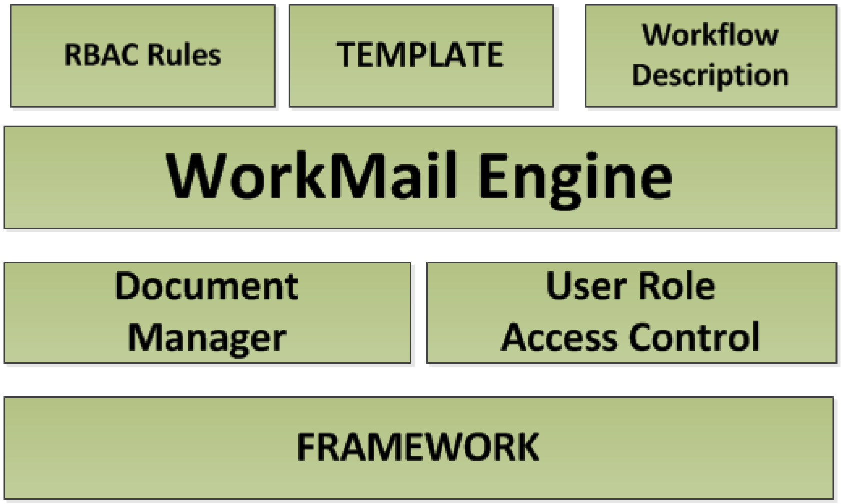


Figure 3 - WorkMail architecture

The core of the architecture is the WorkMail engine that manages all the actions involving the shared document, as, for instance, the definition of the permission set.

The Document Manager component has the functionality of:

- creating template of documents
- manage the life cycle of a single document (create, read, update and delete)

The User Role Access Control allows managing the access to the system according to user and his roles.

The Template is used to adapt the view of the entire document, according with the user' role.

The Access Description Path allows or denies dynamically the access to a particular field of a document basic on:

- user
- user's role
- type of document

- step of document workflow

The “Workflow Description” is a set of rules that describes the document workflow in term of users, roles and document.

Our implementation of above described architecture is the showed in Figure 4

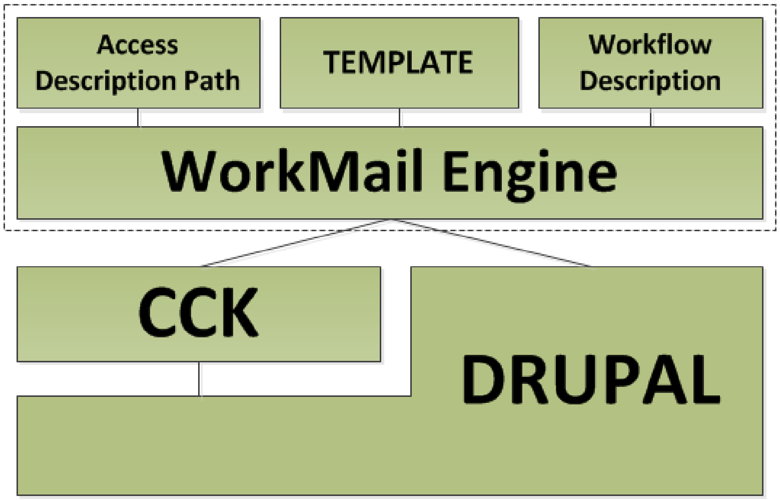


Figure 4 - WorkMail implementation

WorkMail is developed using open source technologies like Apache, PHP and MySQL. -We choose Drupal as content management framework. One of the major advantages of Drupal is its high modularity: it defines hooks, callbacks and API, through which it’s possible to create modules to extend it, leaving unchanged the core.

Moreover, we decide to exploits Drupal also for the implementation of User Role Access System. To implement the Document Manager component we use the Drupal module CCK (Content Construction Kit) [17].

The Engine, the Workflow Description and the Access Description Path have been implemented in the WorkMail module. We developed this module as composed by following three components:

1. Configuration
2. User Interface
3. Engine

Configuration

Due to the flexibility of the CMS Drupal it is very simple to reproduce an organization environment (users, roles, permissions) and especially with the CCK module it’s easy and fast to create any kind of structured document.
With the CCK module we can create content types. Every content type is fully configurable so we can add or remove fields, set the type of these fields and set different permissions per fields in the same document. Every document in the workflow system is a new content type.
Beside fields and content types have own properties, like information about the author and the timestamps of creation and modification.
To make document editing really cooperative different permissions to different roles can be assigned to every single field in the document configuration.

Another feature that comes with Drupal framework is the Taxonomy. In this case taxonomy is used to categorize the email content type. A vocabulary of tags is associated to the content type.

Users and roles are managed by Drupal core. Every user can have one or more roles and the permission setting is connected directly with a role. In this way, if a certain user must perform an action we need to assign a specific permission to a role and then the role to the user.

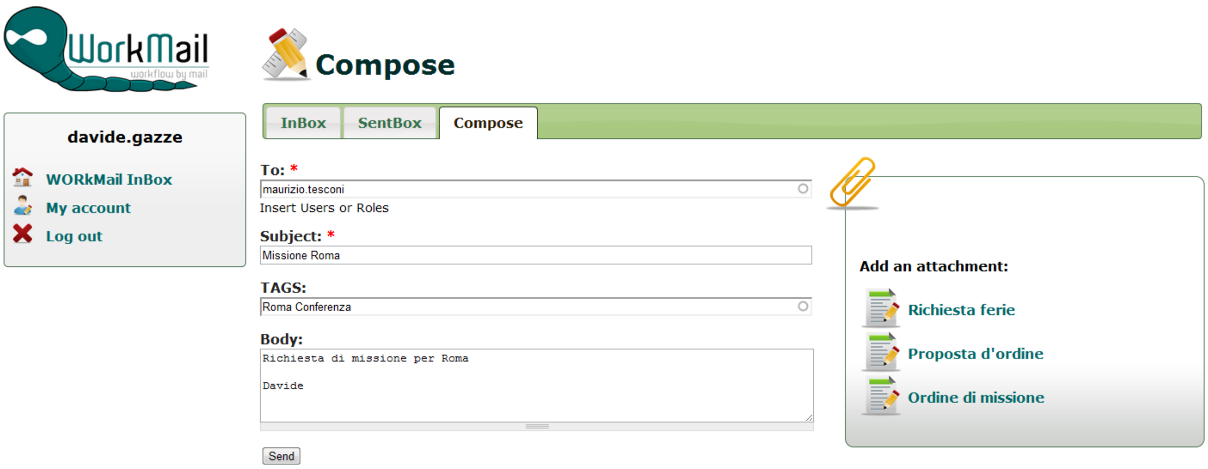
User Interface

While the **Configuration** and **Engine** components regard the administration part, the **User Interface** is look like a web-email client (like Gmail, Yahoo mail, Microsoft mail, etc.).

The User Interface is composed by:

- InBox Tab, that contains the list of all the mail that user has received
- SentBox Tab, whit the list of all the mail that user has sent
- Compose Tab, where the user can create a new mail

While the InBox and SentBox looks like an email client, the Compose Tab is different.



To: *
maurizio.tesconi
Insert Users or Roles

Subject: *
Missione Roma

TAGS:
Roma Conferenza

Body:
Richiesta di missione per Roma

Davide

Figure 5 - Compose message area

As showed in Figure 5, a section containing possible attachments is presented to the user in Compose Tab.

For what concern the “**To**” field, differently from mail, the receiver can be either another user or roles: this can be useful when a document have to reach a group of user with a specific role, for instance, the administrative office. Even if the receiver is a role, the system will resolve the role, in relation with the sending user.

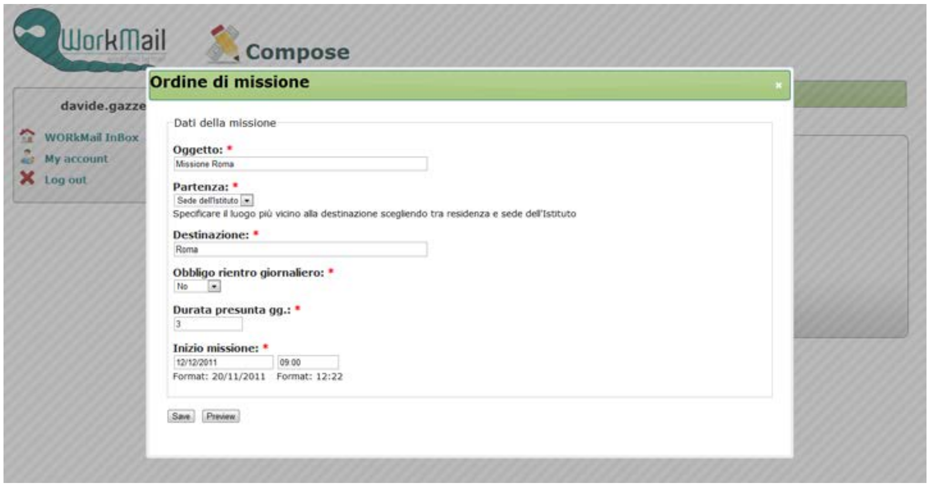
One of the innovative features of WorkMail is the suggestion method. When a user has to send a document to someone, the system suggests her/him a list of possible receivers; this information is stored statically in the system database, at the moment (see future work section for further details). No change will be done to the software, because the employee, in the case of the wrong recipient, could change his mail manually.

This ability simplifies human actions, so a user can decide to follow the proposed flow or change it dynamically.

The fields “**Subject**” and “**Body**” have the same function of a common email client.

Optionally the user can specify one or more tag on the field “**TAGS**” allowing the categorization of emails in folders. For example if a user starts a **Travel Request** (Ordine di missione), he must click on the “Ordine di missione” label.

When a user chooses the type of the attachment, the system shows an instance of the document that the user can fill. The visibility of each field of the document and his proprieties (readable, editable) depend on the user and on the step of the flow. Figure 6 shows an example of T.R. document.



Ordine di missione

Dati della missione

Oggetto: *
Missione Roma

Partenza: *
Sede dell'Istituto
Specificare il luogo più vicino alla destinazione scegliendo tra residenza e sede dell'Istituto

Destinazione: *
Roma

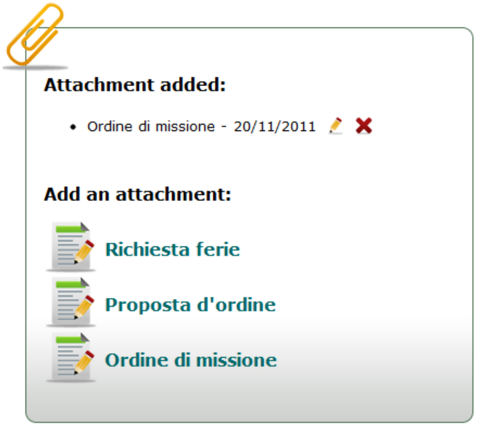
Obbligo rientro giornaliero: *
No

Durata presunta gg.: *
3

Inizio missione: *
12/12/2011 09:00
Format: 20/11/2011 Format: 12:22

Figure 6 - Example of attachment

After saving, the user can see the attachment in the compose area, as showed in Figure 7.



Attachment added:

- Ordine di missione - 20/11/2011

Add an attachment:

Figure 7 - Attachment added

A user can insert one or more attachments.

When the user ends compiling the workmail, he must press the “Send” button and automatically the receiver/s will see a new mail in the InBox.

Engine

The first fundamental concept of the engine is the shared document to fill (the attachment of the workmail). An instance of content type is called node.

The attachment node is unique in the whole document workflow, so users edit the same instance of the document collaboratively; each user edits only its own part of the document.

To implement this feature we need a vessel to deliver the shared document between users, so another content type has been created to contain the attachment.

This new content type (workmail) has as much instances as the number of users involved and performs the role of the classic mail.

The workmail content type contains data about the recipients, the object, the body and other flags indicating if a mail is readed or not.

When a user creates a mail with a document attached the engine creates:

- workmail content types for each user in the recipient
- a single attachment content type

A user creates new workmail content types also when he replies or forwards the mail the attachment.

The second fundamental concept is the permissions set. Obviously, the workmail content type can be seen only by the users in the recipient, but the real innovation is the attachment.

Every attachment node hasn’t a prefixed permission setting, because in our case the flow is dynamic and not static (suggestion can be overwritten). So the engine is capable of calculating the right permission at runtime, depending on the content of the recipient field. We only need to preset generic roles with generic permissions on the document’s field; the engine dynamically associate users to these generic roles during the document flow.

Conclusion and Future Work

In this work, WorkMail has been exposed to present our approach to overcome the limitations of the current document workflow software. The benefit that our model includes is twofold.

From a user's point of view, the use of the email paradigm is helpful to learn using the document workflow system.

From a system's point of view, WorkMail encourages people for collaboration and is exception free; this is allowed by the fact that the user will manage the flow, choosing the receiver of the document at each step. Moreover, we want to underline the advantages using this system as a paperless way to manage documents.

Even if our approach can solve different problems, many technical aspects can be improved and some new features can be developed.

First of all, it will be possible to define an "end" state for the document: this is the case of document that are completely compiled and to extend the search area for filtering a workmail according to sender user, type of attachment, data and tags.

It could be possible to add a feature for digital signature: in this way, documents generated by WorkMail can get legal effect.

Another important feature that can be added is the saving of document in different format, like XML, DOC, PDF, and the integration with the Google Documents' words.

A very interesting and most innovative feature that we are planning to implement is a smart suggestion mechanism. This mechanism will suggest to the user, from each kind of document and for each step of the flow, who could be the receiver of the email.

We will perform a statistical analysis of different information, like involved users, roles, type of attachment and step of the flow, to determine the frequency of the relationship between a sender and receiver, regarding a particular attachment.

Obviously, the engine will need some time to learn the flow and then the recipients to suggest, thanks to that it will be possible to provide a learning mode for first time. In case of significant changes of a flow, it's possible to allow restarting the learning process for specific type of attachment(s).

Bibliography

1. Marchetti, A., Tesconi, M., Minutoli, S. 2005. XFlow: An XML-Based Document-Centric Workflow; *Web Information Systems Engineering – WISE 2005*; 3806, 290-303;
2. Marchetti A., Minutoli S., Lazzareschi P., Martinelli M. A System for Managing Documents in a Step by Step Process. In *Proc. XML World Euro Edition,* March 2001, Amsterdam-Holland.
3. Tesconi M., Marchetti A., Minutoli S., Ronzano F., Web Forms, Technical Report IIT TR-09/2005 April 2005.
4. Krishnan, R.; Munaga L.; Karlapalem K.; XDoC-WFMS: A Framework for Document Centric Workflow Management System. In *Proc. 20th International Conference on Conceptual Modeling – ER 2001*, November 2001
5. L. Aversano, G. Canfora, A. De Lucia, P. Gallucci. Integrating document and workflow management tools using XML and web technologies: a case study. In Proc. Sixth European Conference on Software Maintenance and Reengineering, 11-13 March 2002, Budapest – Hungary, pp. 24-33.
6. P. Ciancarini, R. Tollksdorf, F. Zambonelli. Coordination Middleware for XML-Centric Applications. In Proc. 16th ACM Symposium on Applied Computing – SAC2000, 10-14March 2002, Madrid, Spain.
7. G. Kappel, S. Rausch-Schott, S. Reich, W. Retschitzegger. Hypermedia document and workflow management based on active object-oriented databases. In Proc. of the Thirtieth Hawaii International Conference on System Sciences, Vol. 4, 7-10 Jan. 1997, pp. 377-386.
8. L. Baresi, F. Casati, S. Castano, M.G. Fugini, I. Mirbel, B. Pernici. WIDE Workflow Development Methodology. In Proc. of the International Joint Conference on Work activities Coordination and Collaboration, 1999, San Francisco, California, United States, pp. 19 – 28, ISBN: 1-58113-070-8.
9. F. Casati, M. G. Fugini, I. Mirbel, B. Pernici. WIRES: A Methodology for Developing Workflow Applications. In Requirements Engineering Journal, Springer Verlag Press., Vol. 7, Num. 2, 2002, pp. 73-106.
10. D. Georgakopoulos, H. Hornick and A. Sheth. An Overview of Workflow Management: from Process Modelling to Workflow Automation Infrastructure. In Distributed and

Parallel Database Journal, Kluwer Academic Publishers Press., Vol. 3, Num. 2, April 1995, pp. 119-152.

11. A. Tripathi, T. Ahmed, V. Kakani, S. Jaman. Implementing Distributed Workflow Systems from XML Specifications. Department of Computer Science, University of Minnesota, May 2000. Available at http://www.cs.umn.edu/Ajanta/publications.html
12. R. Tolksdorf, Marc Stauch Using XSL to Coordinate Workflows Kommunikation in Verteilten Systemen 2001 127-138
13. R. Tolksdorf Workspaces: A Web-Based Workflow Management SystemIEEE Internet Computing September 2002 v.6 n.5 p.18-26 Product-focused software process improvement: 8th International Conference on Product Focused Software Process Improvement, PROFES 2007. Jürgen Münch, Pekka Abrahamsson., 2007.
14. Ulrich Kampffmeyer U., *ECM Enterprise Content Management*, Hamburg 2006, ISBN 3936534098. Definitions, Scope, Architecture, Components and ECM-Suites in English, French, and German
15. Stohr, Edward A.,Zhao, J. Leon, 2010. Workflow Automation: Overview and Research Issues.I*nformation Systems Frontiers* 3,281-296
16. Buhler, Paul A.,Vidal, José M.,2005. Towards Adaptive Workflow Enactment Using Multiagent Systems. Information Technology and Management 6, 61-87
17. CCK. Content Construction Kit (CCK). [Online] 2006. http://drupal.org/project/cck.
18. Suchman, 1994; Winograd, 1994; Suchman, 1995, at workflow automation: overview and research issue
19. Joosten S, Aussems G, Duitshof M, Huffmeijer R, Mulder E. An Empirical Study of the Practice of Workflow Management. University of Twente, Center for Tele-informatics, 1994.
20. Mintzberg H. Structure in Fives: Designing Effective Organizations., Englewood Cliffs, N.J.: Prentice-Hall Inc., 1983
21. Abraham Bernstein. 2000. Populating the Specificity Frontier: It-Support for Dynamic Organizational Processes*.* Ph.D. Dissertation. Massachusetts Institute of Technology. AAI0802703.