Johanna: Open Collaborative Technologies for Teleorganizations

Mauro Gaspari  Lucio Picci  Andrea Petrucci
Giovanni Faglioni

Technical Report UBLCS-2002-12
December 2002

Department of Computer Science
University of Bologna
Mura Anteo Zamboni 7
40127 Bologna (Italy)
Recent Titles from the UBLCS Technical Report Series


Johanna: Open Collaborative Technologies for Teleorganizations

Mauro Gaspari1  Lucio Picci2  Andrea Petrucci1  Giovanni Faglioni3

Technical Report UBLCS-2002-12

December 2002

Abstract

The Johanna project develops an open framework to support collaboration in organizations through the integrated use of e-mail and Web services. The Motto of Johanna is “usability and users’ feedback”. Usability is achieved following a non invasive approach: the user exploits e-mail as the main communication tool but he is not constrained to a particular email program, desktop environment and/or application program. In this sense, Johanna supports collaborative work by providing the glue to integrate a set of collaboration components and without requiring a fixed working environment. “User’s feedback” is seen as a key element for success. Too often projects of collaborative environments that were successfully from a technological point a view, failed because users did not materialize. Users’ feedback is guaranteed by the continuous monitoring of users’ needs, as already experimented in a filed test - the first Johanna based site has been delivered to an academic organization. Such a monitoring is carried out on the technology, in order to improve features, interfaces, etc., and, most important, from the point of view of the working of the organization. A meta-language - JML (The Johanna Meta Language) encodes the organizational needs as they are expressed by the users, and allows them to easily configure and customize a Johanna application. Johanna is freely distributed under the GPL licence and runs on the Linux operating system.

In this paper we describe the project’s goals, the design of its architecture, the current status of the project, and its future directions.

1. Dipartimento di Scienze dell’Informazione, University of Bologna, Via Mura Anteo Zamboni, 7, 40127 Bologna, Italy.
2. Dipartimento di Scienze Economiche University of Bologna, Strada Maggiore, 45, Bologna, Italy.
3. giovafaglioni.org
1 Introduction

The Johanna project aims at developing an open framework to support Web services and collaboration technologies in information technologies - rich environments. Its core includes e-mail and Web services as main building blocks to enable the integration of other collaborative technologies. E-mail is the basic tool for communication, and web services provide both human readable information and machine understandable information, according to the principles of the Semantic Web.

Johanna basic philosophy is best described by a simple motto: "usability and users' feedback". Usability is achieved following a non invasive approach: the user exploits e-mail as the main communication tool but he is not constrained to a particular email program, desktop environment and/or application program. In this sense, Johanna supports collaborative work by providing the glue to integrate a set of collaboration components and without requiring a fixed working environment. Thus Johanna differs from tools such as Lotus Notes [3] or Novell Groupwise [6] which constrain the user to adopt a set of groupware (almost always proprietary) tools. "User's feedback" is seen as a key element for success. Too often projects of collaborative environments that were successfully from a technological point a view, failed because users did not materialize. Users' feedback is guaranteed by the continuous monitoring of users' needs, as already experimented in a filed test - the first Johanna based site has been delivered to an academic organization. Such a monitoring is carried out on the technology, in order to improve features, interfaces, etc., and, most important, from the point of view of the working of the organization.

Johanna is a generic framework and can be adopted by a wide range of organizations needing support for e-mail based collaborative work and Web services, provided that they satisfy a set of (well) defined requirements. In this sense, Johanna is not only a software, but also a methodology, that is, a set of principles to obtain given results in terms of organizational change. Johanna is both a methodology for change management, and the software technology to enable such organizational change. The degree of change desired is decided by the "superuser(s)" of Johanna, (a) key figure(s) to be defined below, and can range from a minimum, for very conservative application of the software and of the related methodologies, to the case where a major restructuring of an organization - and of the way in which it relates to its outside environment - is desired.

A high-level language allows the designer(s) to give a high level description (also of a semantic type) of the organization, and of its relations with the outside environment. Such a language effectively describes the working of different organizations and of the desired flows of information, within the organization and between the organization and its environment. The same language also allows for the definition of the desired Web interfaces.

Given this high level description, a compiler generates the glue procedures which integrate the Johanna's main components. These are a Web server, a database containing information on the users and the entities of the application domain, and a mailing-list manager. Both the Web server and the mailing-list manager use the database as a source of information.

Other components, such as notification of relevant messages via mobile telephone, could also be included, but they are not at this stage of the project.

This paper is organized as follows. The next section describes the needs within information technology - rich organized environment that motivate this project. Explicit reference is made to the concept of "teleorganization", which proves to be useful both as a way to spell out the requirements to use Johanna, and to firmly root the project to the available theory of complex organizations, with explicit reference to the role of information technologies within them.

The general architecture of the project is then presented, followed by a description of its basic components and, after that, by a general discussion.

2 Organizations & Teleorganizations: the motivation for Johanna

In order to explain the needs that motivate this project, we are about to provide two different arguments. We start with a bottom-up view, to show that in the "belly" of many organizations some current practices would be carried out more easily by means of collaborative tools of the
type that we propose. After that, we will look at matters from the top, by defining the concept of "teleorganization". We will show that the concept of teleorganization is useful both to highlight the requirements for using Johanna, and also to provide a broader framework for its analysis. The two views, one from the bottom and one from the top, in fact converge to provide a strong motivation for the present project.

2.1 Bottom-up: To Leave or Not To Leave "We do it, Inc."

Let us consider a generic organization where all its members regularly use the Internet for their work. More precisely, over the time they have developed the habit to check their e-mail boxes at least a few times during their work day, and they also regularly access the Web. At least in the industrialized world, both casual observation, and a wealth of available empirical studies, indicate that the set of organizations meeting such requirements - generalized use of e-mail, and use of the Web - is not only growing, but also already quite large as of today.

Typically in these organizations, were there is an expectation that anybody can be reached by e-mail, such messages are used for activating procedures that are part of the functioning of the organization. That is, e-mail is used not just in semi-official contexts, such as within work groups or advisory committees, but also in official communications, and this is true regardless of the presence of digital signatures.

As an example, and as an anticipation to the application that will be described in a further section, at the University of Bologna official Faculty's meetings used to be called by means of closed envelope traditional messages signed by the Dean. Today such meetings are regularly called using email, with no (digital) signature, that is, based on the good faith of the senders and of the receivers, and on a degree of trust on the security of the communication tools that are used.

A big part of modern life, it has been remarked, is today spent in meetings. This piece of popular wisdom reflects the fact that, in any complex organization, there are many organs: boards of directors, committees of various types and degree of officiality, working groups, etc. Within the organizations satisfying the requirements that we have specified, e-mail is today widely used for the purpose of making these organs work.

At a simpler level, it is used for scheduling purposes, as we have seen, sometimes substituting the traditional official letters calling a meeting. Also, it is used for the actual working of many organs: to exchange documents and opinions, and, more and more often, for preparatory work that a "face to face" meeting is called to refine or, sometimes, just to ratify.

Moreover, e-mail is used massively to organize and implement communication with the outside environment. Contacts with vendors, customers, even the press, are carried out more and more in this way.

How do the people actually do all this in real life organizations? Typically, in a very decentralized and inefficient manner. When sending an e-mail message to many people, at a lower level, people use the "cc" (or "carbon copy") field of the header of their email message. More refined users have learned the use of so called "aliases", and group aliases, that allow to do the copying of many addresses once and for all (as long as the list does not change). In both cases, however, different people needing a given list of email addresses, are responsible for setting it up and for its upkeep. A further level of refinement is the use of mailing list software such as Majordomo, Mailman or Listproc, to name a few popular choices. Such software centralize the list of addresses for a given mailing list, thus avoiding the duplications that we have described. Also, they typically provide useful features, such as Web archives, easy management of subscriptions, and monitoring of malfunctioning. These operations are typically carried out by a so called "list master", a person in charge of the upkeep of the mailing list, and typically in power to decide who is in and who is out.

4. In many organizations a distinction between white and blue collar workers is relevant, where blue collar workers do not (yet) satisfy the stated requirement. It can be argued however that a) the information exchange between the two subset of workers is by far less important than the exchange within each of the two subgroups and b) that technologies can be developed to overcome such limitations, for example by printing and physically delivering a message which addressed to a person how does not have an e-mail address, or by posting a message to a hand-held device; such as a SMS to a cellular phone.
However, mailing lists software, while being a very useful tool, are only a partial solution to the problem. Suppose that at “We Do It, Inc.” there are 50 mailing list, and that Mr. Smith, as part of his work duty, is in 30 of them. Now suppose that Mr. Smith leaves his job. He then has to be cancelled by 30 mailing lists, possibly in 30 distinct listmasters.

The solution to this problem in principle is quite simple: “We Do It, Inc.” should put the information about the people who work there, appropriately structured, in a database, and the 50 mailing list should be the result of appropriate queries to that database. When Mr. Smith leaves “We Do It”, he also leaves the data base, and, by such an act, he automatically does not satisfy the query for any mailing list at “We Do It”.

Quite obviously, the same database can be queried in order to publish desired personal information on a Web site, such as a directory of people by office or job.

Although some mailing list managers, for instance Sympa [1], provide facilities to retrieve users information and mailing lists from a database, the integration of the mailing list services with the underlying database is loose. The database is seen just as an external data repository and is not fully integrated with the provided communication facilities. And, vice versa, a database is not aware that the mailing list manager exists.

A better solution can be obtained if a framework where a database is fully integrated with the mailing lists manager would be available. As soon as “We Do It” database schema is designed, all the relations which are necessary to the mailing-list manager should be “automatically” generated, and vice versa the mailing list manager should be (automatically) tailored to the specific “We Do It” scenario.

This is a first description, bottom up, of what Johanna does: organize information in a data base which is fully integrated with a mailing list manager. In Johanna, the design of a database schema automatically generates a connection with a mailing list manager and appropriate queries decide who should receive a given message. Part of the story, then, is about tightly integrating databases, mailing list managers and Web services in a unifying framework. But that is just part of the story, as we will clarify. Before we do move on to the “top down” view, it is useful to consider the matter further: the issue is not just to allow people in organizations to be more efficient, by centralizing several repositories of information, structuring information appropriately and, in this way avoiding duplication of effort and guaranteeing a better upkeep of the information itself.

By organizing information appropriately, the people in the organization are allowed to do things that they could not have done before. In order to see that, let us consider an example, that as a by product allows us to have a glimpse of an organization interacting with its outside environment.

Suppose that Mr Smith does not leave ”We Do It” (maybe because its employer finally decided to adopt Johanna). Suppose that Mr. Smith is in charge of sales, and assume that following the opening of a local branch, “We Do It” decides to offer a special rebate for its products to the people in that area. Suppose further that “We Do It” owns a vast list of e-mail addresses, maybe of people who accessed its Web side and declared some interest.

Mr. Smith contemplates about writing to the people who live in the area of the new branch. If the information about them were not collected for that purpose, it may prove impossible, or very costly, to do it (maybe the right information are there, but are held by Mr Jones, who does not like Mr Smith, and anyway querying the database is not straightforward). If the information are collected within the Johanna framework, Mr Smith, at least in principle, could send the desired message to the appropriate people very easily. 6

The conclusion where we are leading, is that Johanna is not just a way to “rationalize e-mail usage” by means of data warehousing. The idea behind the project is much more general, and it has to do with the construction of a structured repository of the information that is relevant for an ample set of organizations’ procedure, and of the tools to use such information appropri-

---

5. Several databases are supported in sympa for instance Mysql, Postgres and Oracle.
6. By “at least in principle” we mean that it would be easy and that nothing having to do with the technology would stop Mr. Smith from doing it. “Politically” speaking there could be troubles, in case Mr. Jones has control of that part of the dataset. But technology never supersedes political problems, it may just only transform them.
ately. Johanna is then a "machine" to enable relations within the organization, and between the organization and its outside environment.

We have mentioned "an ample set of organizations' procedure", but so far we have come short of declaring that the purpose of Johanna is to describe the whole organization. This is an important issue when we define the domain of application of the project. Clarification of this issue is also accomplished by a "top down" approach, which we now pursue.

2.2 Top down: Teleorganizations

Researchers have long studied the impact of new technologies on organizations. While it can be safely said that the nature of an organization is never neutral to technological change, there is a presumption that change and innovations in the field of information technology are of particular relevance, given that they have an impact on an essential feature of organizations - the flow of information.

Over the last decades, students of organizations have first been interested in the impact of computer technologies within organizations, to later turn their attention to networked computers and, more recently, the Internet.

While there is a wide consensus on the fact that the new information technologies are already shaping a new organizational world, it is not easy to describe any definite pattern, besides a general increment in "interconnexion" among the different atoms of the organizational world.

However, here we make no assumption on the effects of the introduction of Internet technologies on an organization, or on an organization rich environment. We simply describe such a process of technology adoption, using the term "virtualization", that we define as the process by which more and more things are done within a given organization without the need for face to face meetings. We also introduce a terminal point for such a process, by defining the concept of "teleorganization": an organization where all transactions are carried out thanks to Internet technologies.

In this sense, a teleorganization is the end result of a virtualization process, that may or may not be achieved by a given organization. Otherwise, a new organization could be planned from the outset as a teleorganization, with no need for further virtualization.

A teleorganization, from the point of view of information management, is a schema describing a flow of information within an organization, and between an organization and its outside environment. Such information flows are all Internet based: e-mail messages, Web pages, etc., and are implemented, and accessed, in a variety of ways: from desktop computers to personal digital assistants to mobile phones.

What is then Johanna within such a teleorganization "ideal type"? It is the technology that allows such information flows to be effectively managed. Since the description of this flows is a description of a given teleorganization, Johanna represents the organization and its working, and, in a sense, Johanna is the teleorganization.

We have said that a teleorganization is an ideal construct and not, strictly speaking, a description of a given existing organization. Real organizations, at a given point in time, will be a sort of combination of traditional components, where traditional human interactions matter, and of a teleorganization, in the part where information technology mediated interactions prevail. Within real life organization, Johanna, when used, may then represent that part of the organization which is already virtual.

A "virtualization strategy" represents the process by which an organization becomes more similar to a teleorganization. This is what we mean when we observe real life organizations investing in Internet technologies and trying to modify their organizational practices accordingly.

In this context, Johanna, when used, is two things at the same time. Synchronically, it is the set
of technologies that describes the "teleorganizational practices" already in place. Diachronically, it is the set of technologies and practices that embody a given virtualization strategy. The set of practices, not just technologies: behind Johanna there is an idea of how technology enabled change management works.

3 The architecture of Johanna

The Johanna framework exploits a modular architecture. A core, "the Johanna core", provides the glue to integrate other collaborative technologies. The core is based on the integrated use of e-mail and web services and provides a meta-language JML (The Johanna Meta Language) which allows the user to easily configure and customize a Johanna application. The JML uses machine-understandable information such as RDF schema and high level description of interfaces and of the exported services. This infrastructure is the basis for the integration of other tools which support different forms of collaboration (see Section 4).

A general architecture for Johanna is presented in figure 1. The picture shows the role of the JML and the main components of Johanna:

- A web server: this is a generic component which can be instantiated to any web server running on Linux provided that it supports HTTP and PHP extended with SQL. In the actual version of Johanna the web server is apache.
- A mailing-list manager: Sympa [1] which is integrated with several DBMS and authentication procedures.

Moreover, there are a few additional tools (related to the main components) which have been integrated to facilitate the access to the Johanna underlying structure, for instance PhpPgAdmin: an administration tool for Postgres based on PHP.

The JML allows a user to customize the main components of Johanna with the information relevant to a particular application domain.

The first step toward the configuration of a Johanna application is to write an RDF schema which represents the entities involved in an organization. Johanna distinguishes two types of entities:

- Active agents: are the users of the Johanna application. Agents are identified by their e-mail addresses, they are integrated in the Johanna core and thus in the mailing list manager. For example students, professors, secretaries and technicians.
- Passive entities: are all the other relevant entities of the given application domain. The entities which do not communicate among them.

The RDF schema is used to automatically generate the Johanna database schema and relations.

The second step of the configuration concerns the specification of the Johanna interfaces and services. This specification is performed exploiting the JML form language which provides facilities to specify user interfaces based on web forms integrated with the underlying RDF schema.

3.1 JML: the Johanna Meta Language

The aim of the Johanna meta language (JML) is to provide an high-level tool for the Johanna customization and configuration. In the current version of Johanna the JML consists of two languages: RDF schema which is used to specify the database schema and the JML form language which is used to specify user interfaces and web services. In the following two chapters we illustrate these languages by means of a few examples.

9. Sympa runs on Linux and other Unix operating systems and imposes further requirements: the release 5.004_J03 or later of the PERL language, as well as several CPAN modules must be installed. More details can be found in the Sympa user Guide [2].
3.1 A Johanna RDF schema

The first Johanna based site is that of an academic organization. This site has been configured exploiting the JML. The configuration process starts specifying the main agents and entities of the organization. To this purpose JML uses an RDF schema which contains the declarations of RDF classes which represent these entities: students, professors and courses in the Figure 3.1.1.

The next step is the full specification of the above classes. In the JML we assume that each class is specified in a file: “class.name.rdf” which contains the properties that each element of the class should have. In Figure 3.1.1 we present a subset of the properties of the class professor. Note that a professor is an active agent which is automatically added as a user of the mailing list manager. Agents are distinguished from passive entities because the have an e-mail add ress.

An additional file is included in the RDF schema (“start.rdf”) to specify the data types of the properties of each object, the specification concerning the professor class is presented In Figure 3.1.1. For each property we specify the size of the field, if the field is obligatory and if it is a primary key. For example the ID must be always inserted and if is a primary key, while the HopePage and the Phone fields are optional.

All the RDF files we have presented are compiled by RDFtoSQL and the result is the definition of the Johanna database in Postgres SQL. This database is the basis of Johanna and it is used in the next steps of the configuration.

3.1.2 Specifying forms and services

Given a Johanna database the JML provides a facility to specify and automatically generate web forms and services. This is the Johanna JML form language. The specification of a form to input and modify data on professors is presented in Figure 5. Given this specification the JML compiler generates several PHP files which implement the access to the underlying database and the visualization of the forms through a web server. The JML form language allows the user to specify several features to customize the web forms for the particular application. The properties which starts with “input” are those displayed in the form, the other will be used to browse the database.
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdfs:Class rdfs:ID="Professor">
    <rdfs:subClassOf rdfs:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Resource"/>
    <rdfs:label>Professor</rdfs:label>
    <rdfs:comment>The Professors class.</rdfs:comment>
  </rdfs:Class>
  <rdfs:Class rdfs:ID="Students">
    <rdfs:subClassOf rdfs:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Resource"/>
    <rdfs:label>Students</rdfs:label>
    <rdfs:comment>La students class</rdfs:comment>
  </rdfs:Class>
  <rdfs:Class rdfs:ID="Courses">
    <rdfs:subClassOf rdfs:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Resource"/>
    <rdfs:label>Courses</rdfs:label>
    <rdfs:comment>The courses class</rdfs:comment>
  </rdfs:Class>
</rdf:RDF>

Figure 2. The specification of JML classes.

<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdfs:Property rdfs:ID="ID">
    <rdfs:label>ID</rdfs:label>
    <rdfs:comment>Professors identifier.</rdfs:comment>
  </rdfs:Property>
  <rdfs:Property rdfs:ID="Name">
    <rdfs:label>Name</rdfs:label>
    <rdfs:comment>Name of the Professor.</rdfs:comment>
  </rdfs:Property>
  <rdfs:Property rdfs:ID="Surname">
    <rdfs:label>Surname</rdfs:label>
    <rdfs:comment>Surname of the professor.</rdfs:comment>
  </rdfs:Property>
  <rdfs:Property rdfs:ID="Phone">
    <rdfs:label>Phone</rdfs:label>
    <rdfs:comment>Phone number of the professor.</rdfs:comment>
  </rdfs:Property>
  <rdfs:Property rdfs:ID="E_Mail">
    <rdfs:label>E_Mail</rdfs:label>
    <rdfs:comment>E_mail of the Professor.</rdfs:comment>
  </rdfs:Property>
  <rdfs:Property rdfs:ID="HomePage">
    <rdfs:label>HomePage</rdfs:label>
    <rdfs:comment>Home Page of the professor.</rdfs:comment>
  </rdfs:Property>
</rdf:RDF>

Figure 3. The specification of a Professor.
4 Discussion

Johanna is an open system designed to be extended and customized for several application domains. It can be considered open from several points of view:

- The Johanna Meta Language allows a user to specify the structure of an organization in a declarative way. This declarative specification is easy to manage and thus it can be easily extended.

- Johanna exploits e-mail as a main communication mechanism but it does not constrain the user to use a particular electronic mail program. Every e-mail program can be connected to Johanna provided that the user is registered in the main database.

- Other collaborative tools can be integrated in Johanna following three different approaches:
- **tight integration**: which can be achieved extending the JML.
- **functional integration**: new functionalities can be added to Johanna extending the web interface. For examples a tool which presents statistical summaries is presented in Figure 7.
- **loose integration**: which is achieved exploiting the services - exported by Johanna.

- Although also proprietary software can be integrated with Johanna provided that it is conform to the Johanna services interface, all the Johanna components are open source software systems, and the Johanna interfaces are distributed under the GPL agreement themselves.

4.1 **Extending Johanna to the Semantic Web**

The main goal of the next generation web (semantic web) is to support machine executable information enabling intelligent services. One of the main goals of the Johanna project is to develop an infrastructure compatible with the semantic web requirements. Johanna will provide a set of intelligent services characterized by a semantic description. The database will be extended to a knowledge base including data and some explicit knowledge about them. The feature will allow us to extend the set of possible queries to the system enabling intelligent services.

This integration with the semantic web will be achieved extending the JML and adding an ontology server to the set of main Johanna components. The next version of the JML should be able to specify ontologies containing conceptualizations of the application domains, as well as semantic requirements and constraints.

5 **Exploiting Johanna in virtualization processes**

5.1 **Ideas for virtualization processes**

The Johanna methodology for technology enabled change management is a broad framework within defined by a few prescriptions and by two founding principles, both well grounded in the theory of complex organizations.

First, any project of organizational change cannot do without a substantial degree of consensus from the relevant actors of the organizations, and the idea that hierarchical principles can be used to simply “order” a given desired reform is simply wrong: Change management “from above” rarely, if ever, works.
The reason is that organizational change inevitably implies shifts of powers and it affects the utility of the typical actor, and as such it may be face opposition. The actors of an organization typically have many ways to formally respect the received prescriptions for change, while not compiling with the new set of practices that the desired change implies. For every top-down strategy aiming at organizational change, there can be a whole set of bottom-up “counter implementation” practices, against which the hierarchical top can do little. Consensus is key for change: not necessarily everybody has to agree; however, a majority of the most significant actors has to support, or at least not oppose, the proposed changes. Change management is an essentially political activity.

Second, virtualization activities are typically very central to the life of an organization. Internet technologies, by changing the way information flows, almost by definition imply changes in the organization, and reallocation of tasks, power, and utility. Information technologies enabled change management tends to touch the key processes of an organization. For this reason, they are particularly critical.

This state of affairs has a strong implication regarding the definition of the “actors of change”, i.e., those persons, or that person, who strongly support the virtualization strategy and whose job is to deliver it. In principle, it can be one or more people placed in different places within the organization. In fact, we argue, given the centrality of the organizational processes involved, the main actor(s) of change have to be very close to the hierarchical top of the organization. Not only do we need the consensus of the people who are in charge, we need their active involvement in any virtualization strategy in order to be successful. Otherwise, the strategy may not be approved.
in the first place, or, even if it does, the necessary resource-consuming interventions needed to make it successful would not take place.

We then have two conclusions pointing at different directions: on the one hand, a virtualization strategy has to receive consensus from a vast audience within the organization. Also, the “actors of change” have to be very close to the hierarchical top of the organization. The solution to this tension is what we call a “consensus strategy”: an implicit political pact regarding a technology enabled organizational change that involves the vast numbers of actors that either have to be proactive in order to reach given organizational goals, or are anyway in a position to impede the reaching of such goals.

This broad conclusion helps shedding light on the emphasis on users’ needs placed by Johanna. The process of monitoring users’ needs and of managing the feedback of such monitoring into the programming of the Johanna-based interventions, starting from the early stage of planning, is instrumental in achieving their consensus. In order to describe such mechanisms, which together define the Johanna methodology for virtualization processes, we first need to be more precise about the identifications of the actors involved and of their role.

5.2 The Johanna methodology for virtualization processes

The set of indications provided by Johanna for successful Internet technology enabled organizational change start with the definitions of three subsets of figures: the “actors of change”, the “hierarchical top”, and the “relevant actors”.

The actors of change are the people who want to cause an organizational change using (possibly among other things) the Johanna technology. Such people may have sweeping projects of thorough change within the whole organizations. Also, they may have very circumscribed objectives, maybe aiming at modifying just a few organizational practices in some peripheral office.

The (actor(s) of change maybe the person(s) who proposed an organizational change in the first place, maybe stubbornly arguing in favor of it, or, possibly, it could be some “hired gun” who has the task of delivering the desired change. The actors of change maybe just one person, or maybe many, possibly placed in different strategic places within the organization.

We then have the “hierarchical top” figure. In the language of mathematics, such top has to be interpreted "locally": if the project involves just a subset of an organization, the hierarchical top need not coincide with the big boss: it would be (and, typically, it should be) the person(s) who are in charge of the relevant practices within that subset.

The relevant actors are then the people whose consensus on the virtualization strategy has to be obtained, for one of two reasons: because they have to be proactive in order to carry out tasks that are functional for the desired organizational change, and/or they are in a position where they could successfully develop appropriate implementation strategies.

The Johanna methods involves three activities.

First, the actors of change broadly identify the desired goals. This activity has to be carried out with a strong involvement of the hierarchical top, that in principle has to share with the actors of change the responsibility for the success of failure of the process. The actor of change must always be in a position to successfully negotiate with the hierarchical top the necessary actions in order to have a successful virtualization strategy. We call this activity “goal and means identification”.

The second activity is the continuous monitoring of users’ need. Such monitoring starts at an early stage of the project development, so that the project can always be seen as a response to stated needs. The monitoring activity should take place through regular meetings with the relevant actors, both face to face and via e-mail. We call this the “monitoring activity”.

The third activity involves the managing of the feedback. By this we mean on the one hand the feedback from the relevant actors expressed needs, and the (re)definition of the goals and means of the project. Also, managing the feedback involves the necessary political activity able to ensure the necessary consensus for the desired project, both from the hierarchical top and from the relevant actors. We call this activity, which is at the core of the consensus building activity, the “feedback management”.

UBLCS-2002-12
6 Conclusion and Future Developments

The first versions of Johanna have been delivered and used in several academic organizations and projects. Among them a new degree in Internet Economics at the University of Bologna.

The future research will be geared to integrate other collaborative technologies and tools. For instance a chat manager or a sms messenger. An interface which provides automatic query generation, based on semantic context is being implemented. We also plan to extend the Johanna meta language to include more semantics information.

Another important issue concerns the analysis of the organizational needs that are expressed by the users using the analytical tools of the theory of complex organizations, in order to guarantee the necessary consensus to the organizational change that Johanna helps bringing about. We would like to emphasize that Johanna is an example of a technology that enables organizational change, where by “technology” we do not only mean the software, but also the set of codified practices that allow for the planning of the information flow and for the continuous monitoring of users’ need that allows for the continuous feedback process of improvement.

Finally we plan to enhance the Johanna investigating several directions: the integration of a distributed authentication procedure (based on LDAP), the support for user profiles and views, and the addition of new modules, such as a content manager and the integration with a Geographical information system (GIS).

References

REFERENCES


