

## Method and Instruments for Modeling Integrated Knowledge

Olivier Glassey

Fraunhofer Institute for Open Communication Systems (FOKUS), Berlin, Germany  
Swiss Graduate School of Public Administration (IDHEAP), Lausanne, Switzerland  
olivier.glassey@idheap.unil.ch

### 1. Introduction

The goal of this paper is to present a framework developed in order to model and share knowledge within large organizations whether they be private or public. Called MIMIK (Method and Instruments for Modeling Integrated Knowledge), it is based on a methodology, on eight different models for graphical representation and on a knowledge-sharing system. MIMIK provides an explicit representation of reality in terms of concepts (or processes) and their relationships with the overall objective of supporting the processing, storing and retrieving of knowledge.

The field of knowledge representation is not new: mathematical tables, graphs or set theory has been used to structure and represent knowledge long before information systems were developed. However, whereas mathematical models rely only on mathematics for expression, interpretation and validation of a model, other types of models, such as political, social or psychological models need to define their own concepts and relationship types (what is now called an ontology) and furthermore these models have to be validated through experience. Models are used to provide a framework to describe concepts and to reason about these concepts in order to create new knowledge. It is appropriate to note that philosophers have been debating for centuries about knowledge, perception and representation. Although, in this paper we make reference to concepts that are located in other fields of research such as philosophy or psychology, the focus of this paper is on knowledge management within organizations and on knowledge sharing through information technology.

To define our methodology we make use of the knowledge management life-cycle proposed by Nissen et al. (Nissen et al., 2000). They studied four different characterizations of the knowledge management life-cycle and from these they created a combined model consisting of 6 phases: create, organize, formalize, distribute, apply and evolve knowledge. They also presented a review of the tools, technologies and practices were available for each of these phases. In a similar fashion we propose a set of instruments for each phase which are described in section 2.

According to Gordon (Gordon, 2000), there are several accepted methods of knowledge representation in the domain of AI applications and information systems:

- **Rules:** statements which can be proven true within some logical framework, usually represented as a set of attributes value.
- **Frames:** a structural skeleton used to collect information on a given concept and associated actions.
- **Semantic networks:** directed graphs consisting of vertices which represent concepts and edges which represent semantic relations between the concepts.
- **Concept diagrams:** graphical display of concepts connected by directed arcs, in many aspects similar to semantic networks.

The main objective of the MIMIK framework is to support graphical representation, thus we focused on the semantic and conceptual approaches amongst the representation methods listed above. In order to refine this graphical representation approach, we used the Eppler and Burkhard (Eppler, Burkhard, 2004) definition of knowledge visualization. They studied all graphic means that can be used to construct and convey complex insights and they propose several visualization types:

- **Heuristic sketches** or ad-hoc drawings.
- **Conceptual diagrams**: abstract, schematic representations of structural relationships.
- **Visual metaphors**: used to structure information and convey normative knowledge through the connotations of the metaphor.
- **Animations**: interactive descriptions of procedural knowledge.
- **Knowledge maps**: they do not represent knowledge but rather reference it.
- **Scientific charts**: based on computational algorithms.

As MIMIK is a formalized method, we chose not to use ad-hoc drawings or animations. The representation types we found most relevant to our work were the conceptual diagrams and the knowledge maps. In section 3 we will show how MIMIK relies on these conceptual diagrams and knowledge maps to model processes and knowledge within organizations, and particularly those involved with public administration where our approach has been applied and validated.

In order to deepen the definition of conceptual diagrams and knowledge maps, we used Handzic's work. (Handzic, 2004) classifies knowledge representation within organizations in three categories of knowledge maps:

- **Concept-based maps**: taxonomies for capturing and organizing knowledge of an organization around topical areas; they improve visibility and usability of organizational knowledge.
- **Competency-based maps**: overview of expertise that resides in an organization, along with the identification of entities who possess such expertise; they improve the usability of intellectual capital within an organization.
- **Process-based maps**: representation of business processes and related knowledge; they show which type of knowledge is to be applied at a certain process stage and provide pointers to locate that specific knowledge; they are also useful to identify critical knowledge for the core business of an organization.

MIMIK's approach to knowledge representation is based on diagrams that graphically display concepts and their relationships. It does not use rule- or frame-based representation. In order to visualize knowledge we chose to use the knowledge maps approach and we cover the concept, competency and process types of knowledge representation. However MIMIK does not only provide for the representation, organization and storage of knowledge, it also provides instruments that support knowledge sharing, such as the prototype described in section 4.

In their review of the conceptual foundations for knowledge management and in the section dedicated to knowledge transfer, (Alavi, Leidner, 2001) classify knowledge transfer channels as informal or formal, personal or impersonal. They provide a few illustrative examples: coffee break meetings are typically informal, personnel transfers within departments during a training period are formal and personal, knowledge repositories are formal and impersonal, and so on. As a basis for their analytic framework, (Alavi, Leidner, 2001) rely on the four processes of knowledge creation defined by (Nonaka, Takeuchi, 1995):

- From implicit to implicit knowledge: **socialization**.
- From implicit to explicit knowledge: **externalization**.
- From explicit to explicit knowledge: **combination**.
- From explicit to implicit knowledge: **internalization**.

We will analyze how MIMIK supports externalization, combination and internalization of knowledge within groups of people. There are three types of “knowledge communities” which can be found either within an organization or across organizations (CEN, 2004):

- Community of **interest**: a group with mutual interest in a particular topic whose members wish to develop their knowledge in the subject.
- Community of **practice**: people who share insights, expertise and good practices in a specific area of organizations (e.g. finance, human resources, legal departments).
- Community of **purpose**: a team or a group that has to reach specific goals, usually with a shorter time of horizon.

(McLure Wasko, Faraj, 2005) explain that the availability of electronic communication technologies is no guarantee that knowledge sharing will actually take place and examine why people voluntarily contribute knowledge. They make a distinction between communities of practice, i.e. groups whose members are engaged in a shared practice and know each other and work together, and networks of practice, i.e. loosely knit groups of individuals who are engaged in a shared practice but who do not necessarily know each other. (McLure Wasko, Faraj, 2005) studied knowledge sharing within networks of practice and identified two main potential problems:

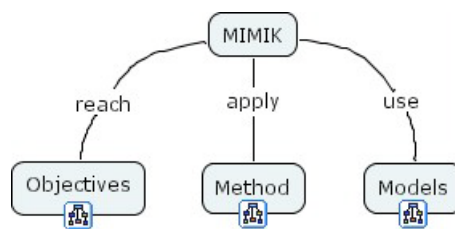
- Knowledge seekers have no control over the respondents and over the **quality** of the responses.
- Knowledge contributors have no assurances that those they are helping will ever help them in return (**reciprocity**).

McLure et al. believe this sharply contrasts with traditional communities of practice where people typically know each other and interact over time, creating expectations of obligations and reciprocity that are enforceable through social sanctions. From their detailed survey on knowledge sharing, they conclude that contributors care just as much about their personal/professional reputation within the network of practice and that reputation is sufficient to guarantee the quality of responses in most cases. Furthermore they found out that contributors do not expect direct reciprocity (from a person that they responded to or helped once) but rather third-party reciprocity. Indeed members expect answers from third-parties, i.e. people that they did not help directly but that were helped by some other member of the community. Third-party reciprocity implies a critical mass of active participants within the network of practice.

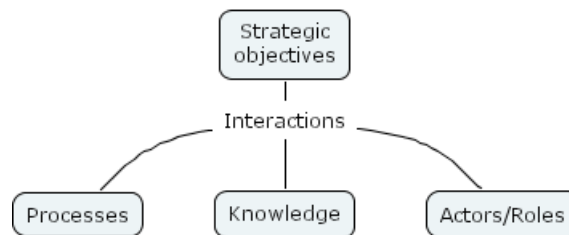
## 2. Modeling Knowledge

As mentioned in the introduction, MIMIK aims at providing a method and instruments for modeling integrated knowledge. **Method** is understood here as **a series of steps** taken to reach a certain objective, and the term **instruments** is used in the sense of concrete or abstract tools intended to assist their users in fulfilling a given objective. These instruments consist mainly of abstract models that represent reality symbolically, making use of concepts and relationships, and implemented as diagrams, i.e. simplified and structured visual

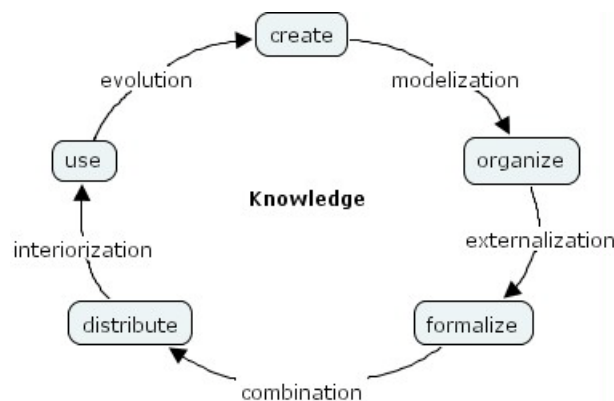
representations of concepts and relations. In this paper we will often use model and diagram synonymously and we will define MIMIK recursively by using MIMIK models.



**Fig. 1.** MIMIK is intended to reach pre-defined objectives by applying a defined method and using a set of existing models.



**Fig. 2.** The objectives of MIMIK are to model the strategic objectives of an organization, to describe processes and knowledge within the organization, to identify actors and roles, and to formalize the interactions between these elements.



**Fig. 3.** The MIMIK method is based on the life-cycle developed by Nissen et al. (Nissen et al., 2000). However we considered that creating and evolving knowledge belonged to the same phase: in most cases organizations do not create knowledge *ex-nihilo* and then work on its evolution; we would rather consider it as a continuous creation cycle. It is also important to note that the steps identified in MIMIK do not necessarily have to be taken in a predefined order.

MIMIK relies on graphical representation to create and organize knowledge and Fig. 3 shows how it supports the knowledge transfer types defined by (Nonaka, Takeuchi, 1995), although these knowledge creation modes are not pure, but highly interdependent and intertwined (Alavi, Leidner, 2001):

- Organizing and formalizing knowledge helps making implicit knowledge explicit (externalization).
- Formalized knowledge can be used to create new explicit knowledge and to share it (combination).

- Shared explicit knowledge can be used and applied by actors that will interiorize this new knowledge (internalization).
- As MIMIK works on knowledge representation and formalization, it is rather self-evident that it does not support implicit to implicit knowledge transfer (socialization).

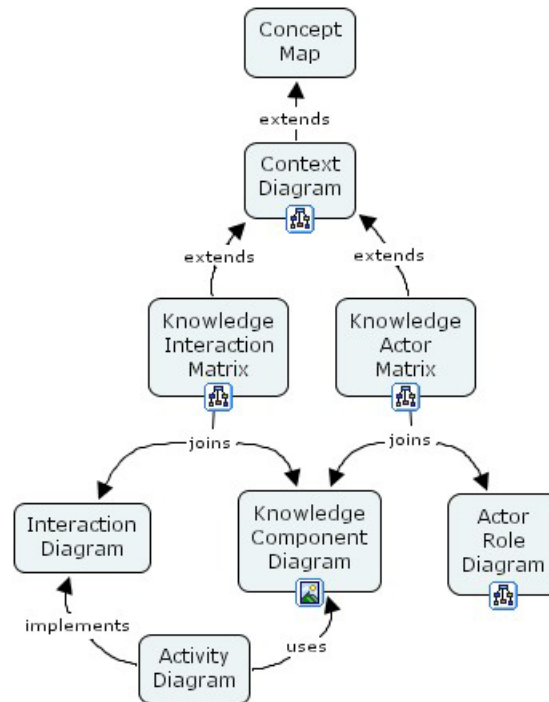
### 3. MIMIK Models

MIMIK is based on the model theory approach developed by (Wyssusek et al., 2001) who aimed at integrating process modeling and knowledge management. They provide an epistemological foundation to justify their approach, but they do not offer any practical methodology or examples. We analyzed the concepts, the models and the application domains of Adonis, OSSAD and UML (Glassey, Chappelet, 2002) and furthermore studied several modeling methodologies with very different backgrounds in order to provide a practical approach:

- Adonis is a software tool for modeling operational processes and it has its own proprietary modeling technique and description language (Junginger et al., 2000). It is one of the many commercial methodologies that are tightly integrated with a modeling environment, such as Aris Toolset (from IDS Scheer AG), Mega Process / Mega Designer (from Mega International, Inc.) or Bonapart (from Pikos GmbH). These tools are widely used in large firms and the public domain.
- ARIS is a very successful method and toolset for modeling business process, used in many universities throughout the world for research and teaching activities (Scheer 2001).
- OSSAD is an open and standard modeling method for organizations and information systems (Chappelet, Snella, 2004; Dumas, Charbonnel, 1990). It was developed as part of an ESPRIT research project. Open source modeling methods are not very common, the only comparable one is OPEN (Object-oriented Process, Environment and Notation), developed and maintained by a not-for-profit consortium (Henderson-Sellers et al., 1998).
- MOKA is a methodology for developing knowledge-based engineering applications, in particular in the fields of aeronautical and automotive industries, and it is used for designing complex mechanical products (MML, 2000).
- CommonKADS is a methodology developed to support structured knowledge engineering also as part of the ESPRIT Program. It is now essentially a European *de facto* standard for knowledge analysis and knowledge-intensive system development, used by many companies and universities (Schreiber et al., 2000).
- UML is a non-proprietary modeling and specification language, with a strong focus on object-oriented software-intensive systems (Booch et al., 1999).

We also used the work of Mentzas et al. (Mentzas et al., 2001) on process modeling and workflow models. They present several techniques for workflow modeling, amongst them activity-based workflow modeling. A workflow is defined as a partial or total order of a set of tasks, and a task is defined as partial or total orders of operation or descriptions for human actions. Mentzas et al. also integrate the concepts of manipulated objects (documents, data records, hardware, etc.), roles (responsibility for a particular task) and agents (humans or information systems filling roles).

MIMIK consists of 8 types of diagrams (Fig. 4), most of them being inspired or directly taken from existing modeling techniques. As in UML or other modeling languages, it is not necessary to use all of them in order to provide a good representation of reality. Users should rather select the diagrams that suit their needs and goals in terms of modeling.



**Fig. 4.** Metamodel showing formal relations between the different types of diagrams proposed by the MIMIK framework.

Concept maps are the top-level diagrams that show the strategic goals of an organization in terms of functions or processes and their relationships. The metamodel of our framework is itself described with concept maps (Fig. 1-4). These concept maps can be decomposed into several levels. When a concept map does not have any “child” map, it is considered as a terminal node. Normally a terminal node of a concept map should be described by a context diagram. Concept maps were developed by (Novak, 1998) and we designed them with IHMC CmapTools (cmap.ihmc.us).



**Fig. 5.** Context diagrams are almost exactly the same as use cases in UML, with the addition of the concept of knowledge packet.

A knowledge packet is an abstract representation of a set of knowledge components. These components encapsulate documents, databases, files, implicit knowledge, etc. and they will be explained in detail further on in this section. Knowledge packets represent, at the abstract level, what type of knowledge is necessary in order to complete a process and which knowledge is relevant in a given context. At this level actors are also represented generically and no difference is made between physical actors and their roles. This will be described with the actors-roles diagrams, as explained further on.

One of the goals of MIMIK is to describe the interactions between strategic goals, actors, knowledge and processes. That is why we developed the concept of knowledge matrices, in order to provide formal links between knowledge and processes (Knowledge-interaction matrix) and knowledge and actors (Knowledge-actor matrix). Each context diagram has a

single corresponding knowledge-interaction matrix and a single corresponding knowledge-actor matrix.

	Model objectives	Model processes	Model knowledge	Model actors/roles
Cmap	✗			
Cdiag	✗		✗	✗
Kmax		✗ join	✗ join	
ARD				✗
Kcomp			✗	
Idiag		✗		

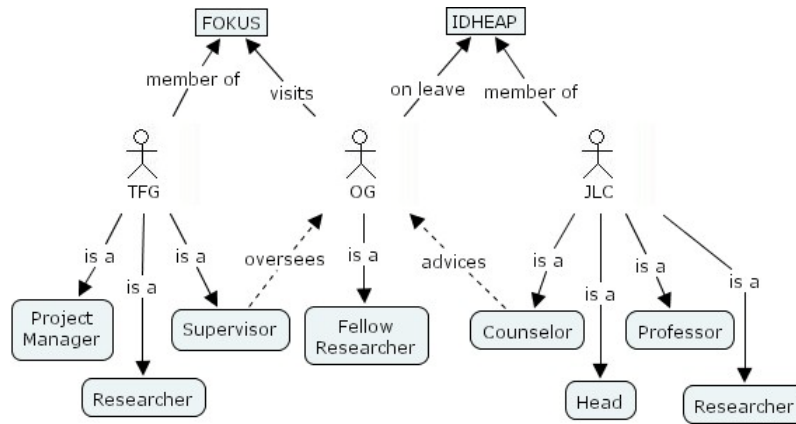
**Fig. 6.** Knowledge-interaction matrix formally linking MIMIK diagrams (represented as knowledge components) to MIMIK strategic processes or objectives. Some new models are introduced here, but they will be clarified further on.

Fig. 6 demonstrates how diagrams are applied to reach given objectives within MIMIK. Concept maps (*Cmap*) and context diagrams (*Cdiag*) are used to model the strategic objectives of an organization. *Cdiag* model knowledge at the abstract level and they also integrate actors, thus showing the context in which knowledge is applied. Interaction diagrams (*Idiag*) provide descriptions for the processes. Knowledge component diagrams (*Kcomp*) provide a symbolic representation of knowledge. As their name implies, actors-roles diagrams (*ARD*) show actors and their roles. Finally the knowledge matrices (*Kmax*) provide a link between knowledge and processes, and knowledge and actors. Fig. 7 shows where MIMIK diagrams are used within the MIMIK methodology.

	create	organize	formalize	distribute	use
Cmap	✗				
Cdiag		✗			
Kmax		✗			
Idiag			✗		
Kcomp			✗		
KnowS				✗	✗

**Fig. 7.** Knowledge-interaction matrix formally linking MIMIK diagrams (represented as knowledge components) to MIMIK methodological steps. KnowS is the prototype of knowledge sharing we developed (see section 4).

Actor-role diagrams can be either classical organizational charts, actors-role diagrams such as shown in Fig. 8, or matrices that formally link actors and roles in cases where the organization is too complex to be able to show these relationships graphically in an intelligible way.



Note – correct spelling of advises

**Fig. 8.** Actors-role diagrams showing administrative units, actors and their roles within the development of MIMIK.

In a similar fashion to the way that knowledge-interaction matrices link knowledge components and interactions, knowledge-actor matrices (Fig. 9) create a formal relation between knowledge components and real actors within an organization. This proves very useful in order to introduce implicit knowledge in a graphical model: it might not be possible to transform such implicit knowledge into explicit knowledge but at least we know who has this knowledge and where it is in the organization.

	OG	JLC	TFG
UML	fluent		fluent
OPRL	fluent	expert	
LKS	basic		expert
RSS	fluent		basic

**Fig. 9.** Knowledge-actor matrix showing who has knowledge of UML, OPRL (Office Process Redesign Language, an evolution of OSSAD), Legal Knowledge System (LKS) and RSS (Real Simple Syndication, based on XML and used for our prototype), and what level of knowledge they have. These were the main approaches (methodologies) that were used as a basis for the development of MIMIK.

As the main goal of our work was to represent knowledge graphically, we needed a formalism to do so and we analyzed what was being done in process methodologies. One basic way to represent knowledge in organizations is the use of business rules (introduced in Brachman, Levesque, 1985 and popularized by Ross, 1997). These business rules can be used to represent all types of activity and do not have to be implemented in an information system.

Some of these rules are implicit, meaning that they are not written anywhere, they are an integral part of the “business culture”. However the basic formalism proposed by (Ross, 1997) is not sufficient in all cases to model knowledge. Indeed, a “knowledge unit” (Fraser et al., 2003) is anything worth storing that may help things to be done better in the future: help, best practices guidelines, examples, stories, lessons learned, troubleshooting advice or training material and business rules cannot model all these types of knowledge. A different approach is described by authors such as (Gamper et al., 1999) and (Gruber, 1993) that use ontologies (explicit specification of a conceptualization, the latter consisting of identified concepts and relationships assumed to exist and to be relevant) in order to model knowledge. We prefer this method as we previously used RDF to build a data-model for e-Government (Glasse, 2004) and found it more powerful and flexible than classical data models such as Entity-Relationship-Model used in ARIS (Scheer, 2001) or than business rules. RDF (Resource Description Framework) is a W3C standard for defining metadata and encoding machine-readable semantics (Noy & al., 2000). It is based on XML and uses graph theory to represent knowledge. It is also a suitable format for specific domain ontology modeling.

However ontologies still cannot represent complex knowledge such as storytelling or human advice. As Samuel Johnson stated in the 18th century: “Knowledge is of two kinds. We know a subject ourselves or we know where we can find information upon it”. The goal of the component-based architecture we propose is to model “the information upon knowledge”, i.e. to provide metadata on “knowledge units”. We identified several attributes that allowed us to do so, beginning with the type of knowledge. (Capurro, 2004) proposes a knowledge typology by comparing the one defined by (Zahn et al., 2000) with the classical Aristotelian one. He furthermore builds on existing knowledge typology, such as the one defined by Gilbert Ryle that distinguishes between know-how and know-that. Here we will only summarize the main points of Capurro’s knowledge typology:

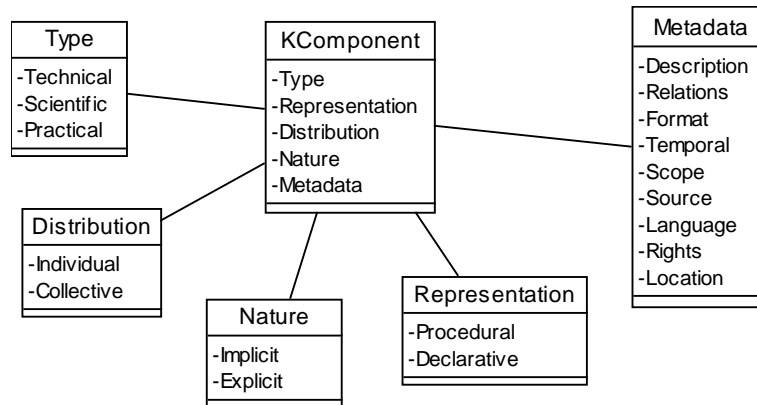
- Know-how: knowledge about how to make things (*technical* knowledge) and knowledge acquired through experience and remembrance (empirical knowledge).
- Know-why: logical reasoning (*scientific* knowledge).
- Know-what: knowledge about the best means to achieve given goals, usually a combination of know-how and know-why (*practical* knowledge).

(Capurro, 2004) furthermore states that what can be managed is information or explicit knowledge and that implicit knowledge can only be “enabled”. In this context, *explicit* means that it can be clearly observed and expressed (and also digitized), as opposed to *implicit* knowledge that cannot be directly formulated (skills, experiences, insight, intuition, judgment, etc.) When knowledge is explicit, it can be represented as declarative or procedural knowledge. We are aware that in the domain of cognitive sciences, the distinction between procedural and declarative models is related to the brain memory system (see for example Ullman, 2001), but in this paper we used these terms in a limited sense, as defined in the computer science and AI literature (see for example):

- *Declarative* knowledge components represent domain knowledge (facts, events, etc.) in terms of concepts and relations.
- *Procedural* knowledge components describe actions to be taken in order to solve a problem step by step.

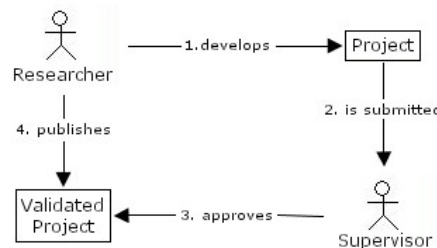
For cases where knowledge is implicit and cannot be formalized, we introduced the concept of *distribution*: knowledge can be *individual* or *collective*, and in both cases components identify who has this knowledge or where it can be found. Finally we added a set of *metadata*

(know-where, know-when, know-who, etc.) describing these knowledge-components and making it possible to manage them.



**Fig. 10.** Knowledge component architecture shown as a class diagram, but it can also be formalized in RDF.

In order to represent processes, MIMIK uses standard UML collaboration and sequence diagrams, grouped under the name of interaction diagrams (*Idiag*), as well as UML activity diagrams. In UML an interaction is the specification of how messages are sent between objects or other instances, and interaction diagrams (sequence or collaboration diagrams) emphasize object interactions. We comply with this definition and use collaboration or sequence diagrams to specifically describe each interaction shown in knowledge-interaction matrices. In (Glasse, Chappelet, 2002) we have shown these diagrams are equivalent to those used at the operational level by specialized process modeling techniques such as OSSAD or Adonis. For MIMIK we chose UML as it is a *de facto* standard, but other types of activity or workflow based diagrams would also be suitable. As we studied process modeling in detail in the aforementioned publication, we will not go into details here, but we will show one example of a collaboration diagram. In UML collaboration and sequence diagrams are isomorphic, meaning that they can be transformed from one to another without loss of information. Sequence diagrams have a temporal focus and collaboration diagrams a structural one. Finally knowledge packets and components (*Kcomp*) are integrated as objects (or instances of the *Kcomp* class) in interaction and activity diagrams, whereas the roles shown in collaboration or sequence diagrams are those who were defined in actor-role diagrams.



**Fig. 11.** Simplified collaboration diagram showing the process model for developing a research project and publishing it.

At the operational level, activity diagrams represent processes with a focus on control and information flows. Each process is viewed as transforming a set of inputs, modeled by incoming flows, into outputs or outgoing flows (Lee, Winer, 2003): elementary activities or operations are performed in a chronological manner, swimlanes can be used to show which actors or roles are responsible for elementary activities or operations. Conditions, parallel

operation and start/stop points are used to control the execution flow of the elementary activities or operations.

#### 4. Prototype

The prototype of a knowledge-sharing system (*KnowS*) is based on RSS (Really Simple Syndication), a family of XML file formats for web syndication. There are two main branches in the RSS family: RSS 1.0 is based on RDF and RSS 2.0 uses an explicit extension mechanism based on XML namespaces. These technical subtleties are not taken into account in this paper, as there are tools that allow compatibility between RSS 1.0 and 2.0. What is relevant is that the RDF-based knowledge components presented in the previous section can be described in RSS. Indeed the XML files (or RSS feeds) provide “items” containing short descriptions of web content together with a link to the full version of the content. In order to access these feeds, users rely on applications called feed readers that check RSS-enabled Web pages and retrieve any updated content that it finds.

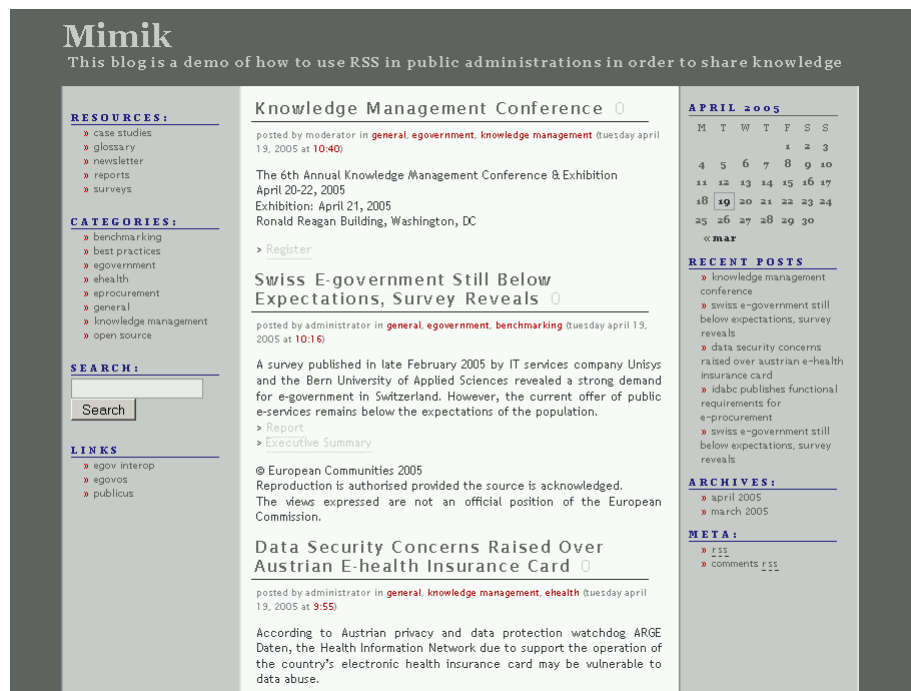
Websites featuring RSS feeds include The New York Times, The Wall Street Journal, BBC, news.com, Liberation, etc. RSS is widely implemented in the weblog community in order to share the latest weblog entries. According to a Pew Internet and American Life Project survey (Rainie, 2005), there were 8 millions bloggers in the United States at the beginning of 2005 and 27% of Internet users say they read blogs. Furthermore (Gordon, 2003) showed that RSS can be used for public participation platforms, for example to facilitate public consultation, deliberation, and participation or “engagement” in policy-making processes such as urban planning. For more on RSS we recommend (Winer, 2005) or wikipedia.org.

Using blog platforms, end users can publish new knowledge via either a Web interface, a simple email sent to a special address or a dedicated feed publishing client. This requires no specific knowledge (other than being able to send an email, at the most basic level of use), the input text is automatically transformed in an RSS feed by the system. A moderator or domain expert should validate this new content before it is made available to anyone, but it is not required. Furthermore, specific thematic RSS feeds can be defined: users can then choose precisely what knowledge they want to receive. Once new knowledge has been published, it can be used in very flexible ways. Users can simply visit the Web page of the blog, but they can also use Web aggregators such as Bloglines.com, their own email client or a specialized feed-reader that provides more advanced functionalities. However, the Internet and American Life Project survey on blogs (Rainie, 2005) stated that only 5% of Internet users rely on dedicated aggregators to get RSS feeds. RSS aggregators and, to some extent, email clients offer powerful content management capabilities, such as filters to limit access to relevant content: a user can for example subscribe only to feeds that aggregate content on given issues and furthermore filter content. This is very useful to avoid information overload, that is to limit the risk that the users will not read the feeds anymore because they receive too much irrelevant information.

Finally RSS feeds support “enclosures”, which allow the addition of any type of multimedia files, similar to an attachment in an email. Thus we added this functionality to our prototype: it can be used to share automatically new documents, files or any piece of digitalized information. With an advanced RSS reader, it becomes possible to check periodically (once a day, every week, etc.) selected feeds and to download relevant documents automatically.

The KnowS prototype comes in three “flavors”, there are three technical architectures corresponding to different needs. The first one is based on existing services: Blogger.com, a

weblog platform owned by Google, and Feedburner.com, a free post-processing service that allows publishers to enhance their feeds. It allowed us to create a simple proof of concept for what was presented in the previous paragraphs, without any material acquisition or software installation. However Blogger does not support categories for blog entries: we believed that this could prove very useful to organize and personalize knowledge. Indeed by matching categories and interactions such as defined in the knowledge-interaction matrices, we could implement a system corresponding to the knowledge models created with MIMIK. Moreover Blogger's user rights management is very limited. For these reasons, we built a second KnowS prototype (Fig. 12) operating on a dedicated blogging platform: WordPress offers good categories' management tools and 10 levels of users' rights (read, write, edit, validate, publish...) and is available at wordpress.org.



**Fig. 12.** Web interface to the Mimik demo blog.

We consider that this type of blog publishing platform, powerful and simple to use, would be sufficient in most cases in order to implement a knowledge-sharing system for public organizations. In some complex cases however, a complete content management system supporting RSS would probably be necessary. In order to illustrate this type of technical architecture, we created a third prototype based on Agora, a specialized Web content platform developed on top of the SPIP system (www.spip.net) at the initiative of the French government (www.agora.gouv.fr). To summarize the advantages of this system, it allowed us to integrate categories and keywords, thus to have a finer knowledge organization, and to assign the control over the content of single categories and keywords to given users. With these functionalities we were able to match not only the knowledge-interaction matrices in the system, but also the knowledge-actor matrices. Finally, let us mention that the two more complex versions of KnowS integrate directly most metadata defined for the knowledge components. RSS provides a description for content and integrates various tags (author/source of an RSS element, publication date, language). The KnowS prototype supports several temporal markers (published, updated, validity), it has advanced rights management functions and supports embedded files with MIME type description (format). Moreover it is possible to use permalinks (a type of URL designed to refer to a specific information item and to remain unchanged permanently) to identify the location of a knowledge element and to show

relations between these knowledge elements with track-backs (system allowing bloggers to see who has written another entry concerning a given post).

## 5. Conclusions

As we mentioned in the introduction, the MIMIK framework is focused on knowledge representation and furthermore provides instruments for knowledge creation and transfer. Throughout the paper we presented MIMIK, its models, and the KnowS prototype. We will now summarize how they relate to the knowledge typology presented in section 3 (Capurro, 2004):

- **Know-how** is represented with process models, concept maps at the strategic level and interaction/activity diagrams at the operational level.
- **Know-why** is described with knowledge packets at the abstract level and knowledge component at the operational level.
- **Know-what** is represented by context diagrams at the abstract level and knowledge interaction matrices at the operational level.
- In addition MIMIK is able to represent **who-does-what** with actor-role diagrams and the **who-knows-what** with knowledge actor matrices.

In section 2 we said that MIMIK method supports knowledge creation and transfer, but we did not explain that claim in detail, as we had not yet described the various MIMIK models. We will briefly recapitulate here how we think they support knowledge transfer types such as defined by (Nonaka, Takeuchi, 1995):

- **Externalization:** the eight types of MIMIK diagrams (see Fig. 4) can be used to organize and formalize knowledge, thus to make implicit knowledge explicit.
- **Combination:** the use of MIMIK diagrams, and particularly knowledge matrices, and of the KnowS knowledge sharing prototype support creation and transfer of new explicit knowledge.
- **Internalization:** users of the KnowS prototype have facilitated access to relevant knowledge that they will be able to apply in their work.

In this concluding section, we should also state explicitly how MIMIK models are related to the KnowS prototype features:

- KnowS categories are used to describe MIMIK interactions.
- MIMIK activities can be linked to keywords or tags.
- Knowledge components are described with RSS posts and can (if they are digitized) be integrated to RSS feeds as enclosures.
- KnowS users should assign relevant categories to each post in order to reflect the knowledge interaction matrices.
- Categories and keywords can be assigned to KnowS users in order to reflect the knowledge actor matrices.

In the introduction we also discussed several types of knowledge visualization: concept maps can be used to create heuristic sketches (although they are conceptual diagrams); context diagrams, interaction diagrams and knowledge components diagrams are typical conceptual diagrams; knowledge matrices provide basic knowledge maps. MIMIK also provides the advantages of several knowledge representation approaches: description of organizational knowledge at the abstract level with concept maps; an overview of knowledge in context with

knowledge-interactions matrices and identification of knowledge possessors with knowledge-actors matrices; a description of the core business of an organization with process models. This combination of knowledge representation categories is formalized in the metamodel of MIMK.

Finally, in the introduction we briefly explained some issues related to knowledge sharing within networks of practice, but we did not investigate the problems of control over knowledge quality and of reciprocity. We plan to do this in future work, by testing the KnowS prototype on a larger scale. At this point we only demonstrated that it was possible to define a technical architecture supporting knowledge creation and transfer. As the MIMIK models at the operational level rely on standards such as UML and RDF, they can be used to develop information systems: code generation in Java or any other programming language, creation of XML files that can be transferred between various applications, etc.

### Acknowledgement

This work was supported by a fellowship of the Swiss National Science Foundation (PA001-101413) during my stay at the Fraunhofer FOKUS Institute in Berlin. I also wish to thank Prof. Jean-Loup Chappelet (Swiss Graduate School of Public Administration) and Dr. Thomas Gordon (FOKUS), who helped and inspired me during this project.

### References

- Alavi M, Leidner DE. 2001. Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS Quarterly* **25**(1): 107-136.
- Brachmann, RJ, Levesque, HJ (Eds). 1985. *Readings in Knowledge Representation*. CA: Morgan Kaufmann.
- Booch G, Rumbaugh J, Jacobson I. 1999. *The Unified Modeling Language User Guide*; MA: Addison-Wesley.
- Capurro R. 2004. Skeptical Knowledge Management. In *Knowledge Management: Libraries and Librarians Taking Up The Challenge*, Hobohm HC (ed.); Munich: Saur; 47-57.
- CEN. 2004. European Guide to good Practice in Knowledge Management. <http://www.cenorm.be/cenorm/businessdomains/businessdomains/iss/cwa/knowledge+management.asp> [25 May 2005]
- Chappelet JL, Snella JJ. 2004. *Un langage pour l'organisation : l'approche OSSAD, 3rd Edition*; Lausanne: Presses Polytechniques et Universitaires Romandes.
- Dumas P, Charbonnel, G. 1990. *La Méthode OSSAD, pour maîtriser les technologies de l'information*; Paris: Les Editions d'Organisation.
- Eppler MJ, Burkhard RA. 2004. Knowledge Visualization. <http://www.knowledgemia.org/modules/pub/view.php/knowledgemia-67> [21 April 2005]
- Fraser J, Adams N, Macintosh A, McKay-Hubbard A, Pariente Lobo T, Fernandez Pardo P. 2003. Knowledge Management Applied to E-government Services: The Use of an Ontology. In *Proc. 4th International Working Conference KMGov 2003*, Wimmer, M. (ed.); Heidelberg: Springer Verlag; 116-126.
- Gamper J, Nejdil W, Wolpers M. 1999. Combining Ontologies and Terminologies in Information Systems. In *Proc. 5th International Congress on Terminology and Knowledge Engineering*, Innsbruck, Austria.
- Glasse O, Chappelet JL. 2002. Comparaison de trois techniques de modélisation de processus: ADONIS, OSSAD et UML; Working paper; Lausanne: IDHEAP.

- Glasse O. 2004. Developing a one-stop government data model. *Government Information Quarterly* **21**(2), 156-169.
- Gordon JL. 2000. Creating Knowledge Maps by Exploiting Dependent Relationships. *Knowledge Based Systems* **13**; 71-79.
- Gordon TF. 2003. An Open, Scalable and Distributed Platform for Public Discourse, *Informatik 2003* **2**, 232-234.
- Gruber T. 1993. Towards principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies* **43**(5-6), 907-928.
- Handzic M. 2004. The Role of Knowledge Mapping in Electronic Government. In *Proc. 5th International Working Conference KMGov 2004*, Wimmer, M. (ed.); Heidelberg: Springer Verlag; 9-17.
- Henderson-Sellers B, Simons T, Younessi H. 1998. *The OPEN Toolbox of Techniques*; MA: Addison-Wesley.
- Junginger S, Kühn H, Strobl R, Karagiannis D. 2000. Ein Geschäftsprozessmanagement-Werkzeug der nächsten Generation - ADONIS: Konzeption und Anwendungen, *Wirtschaftsinformatik*, **42**(5), 392-401.
- Lee J, Wyner GM. 2003. Defining specialization for dataflow diagrams. *Information Systems* **28**(6); 651-671.
- McLure Wasko M, Faraj S. 2005. Why should I share? Examining Social Capital and Knowledge Contribution in Electronic Networks of Practice. *MIS Quarterly* **29**(1), 35-57.
- MML Working Group. 2000. MOKA User Guide - MOKA Modeling Language Core Definition; MOKA Project - ESPRIT 25418, Deliverable D1.3; <http://web1.eng.coventry.ac.uk/moka/mml.htm> [15 September 2004]
- Nissen ME, Kamel MN., Sengupta, KC. 2000. A Framework for Integrating Knowledge Process and System Design. *Information Strategy: The Executive's Journal* **16**(4); 17-26.
- Nonaka I, Takeuchi H. 1995. *The Knowledge Creating Company: How Japanese Companies Create the Dynamics of Evolution*; Oxford University Press.
- Novak JD. 1998. *Learning, Creating , and Using Knowledge: Concept maps as facilitative tools for schools and corporations*; Mahwah, N.J.: Lawrence Erlbaum & Assoc.
- Noy F, Ferguson RW, Musen M. 2000. The Knowledge Model of Protégé-2000: combining interoperability and flexibility. <http://www.smi.stanford.edu/projects/protege/> [12 January 2003]
- Rainie L. 2005. The State of Blogging, Pew Internet & American Life Project, February 2, 2005. <http://www.pewinternet.org/> [10 March 2005]
- Ross R. 1997. *The Business Rule Book: Classifying, Defining and Modeling Rules, Version 4.0*; Houston: Business Rules Solutions.
- Scheer AW. 2001. *ARIS - Modellierungsmethoden, Metamodelle, Anwendungen*; Berlin Heidelberg: Springer-Verlag.
- Schreiber G, Akkermans H, Anjewierden A, de Hoog R, Shadbolt N, Van de Velde W, Wielinga B. 2000. *Knowledge Engineering and Management: The CommonKADS Methodology*; Cambridge: MIT Press.
- Ullman MT. 2001. A Neurocognitive Perspective on Language: The Declarative/Procedural Model. *Nature Reviews - Neuroscience* **2**(10), 717-726.
- Winer D. 2005. RSS 2.0 Specification, RSS at Harvard Law. <http://blogs.law.harvard.edu/tech/rss> [10 March 2005]
- Wyssusek B, Schwartz M, Kremberg B, Baier F, Kralmann H. 2001. Business Process Modeling as an Element of Knowledge Management - A Model Theory Approach. In *Proc. Managing Knowledge 2001: Conversations and Critiques*. Leicester University.

Zahn E, Foschiani S, Tilebein M. 2000. Nachhaltige Wettbewerbsvorteile durch Wissensmanagement. In *Wettbewerbsvorteile durch Wissensmanagement*, Krallman H (ed.); Stuttgart: Methodik und Anwendungen des Knowledge Management; 239-270.