ŽINIŲ VADYBA

From graphical modelling to cross-organizational knowledge-sharing in the public sector

Olivier Glassey

Swiss Graduate School of Public Administration,
Lausanne, Switzerland

Šveicarijos aukštoji viešojo administravimo mokykla
Rozana, Šveicarija
Fraunhofer FOKUS, Berlin, Germany
E-mail: oglassey@yahoo.fr

This paper describes a framework that supports knowledge modelling and sharing within public administrations and across organizations, as well as a prototype of such a knowledge-sharing system. We will first give a brief theoretical introduction on processes of knowledge creation, transfer and application, and then we will present our framework and discuss how it relates to these processes. We will furthermore illustrate this with the architecture of the prototype we developed in this context.

The goal of this paper is to describe a framework we developed in order to model and share knowledge within public administrations and across organizations. This framework is based on a formal methodology, on eight different models for graphical representation and on a knowledge-sharing system.

To define our methodology we used the knowledge management life-cycle proposed by Nissen, Kamel & Sengupta (2000). They studied four life-cycle representations and created an amalgamated model consisting of 6 phases: create, organize, formalize, distribute, apply and evolve knowledge. They made a review on what tools, technologies and practices were available for each of these phases. Likewise we propose a set of instruments for each phase, which we will describe in section 2.

The focus of our framework is on graphical representation: Eppler & Burkhard (2004) define knowledge visualization as all graphic means that can be used to construct and convey complex insights. 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

In their review of 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 mention a few 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. For their analysis 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*

One of the focuses of this paper is knowledge sharing across organizations; we therefore propose an architecture for such a system and developed a small prototype (section 3). This architecture is not only technical, but also organizational, as 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 identify the main problems that sharing knowledge could arise in what they call 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:

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

From their detailed survey on knowledge sharing, McLure Wasko & Faraj (2005) conclude that contributors care about their personal/professional reputation within the network of practice and that reputation is a sufficient mean to guarantee the quality of responses in most cases. Furthermore they found out that contributors do not expect direct reciprocity but rather third-party reciprocity, given that there is a critical mass of active participants within the network of practice.

Knowledge modelling

We developed a framework called MIMIK (Method and Instruments to Model Integrated Knowledge): its supports the 6 phases developed by Nissen & al. (2000): *create*, *organize*, *formalize*, *distribute*, *apply* and *evolve* knowledge. However we considered that creating and evolving knowledge belonged to the same phase: in most cases organizations do no create knowledge ex-nihilo and then evolve it; we would rather consider it as a continuous

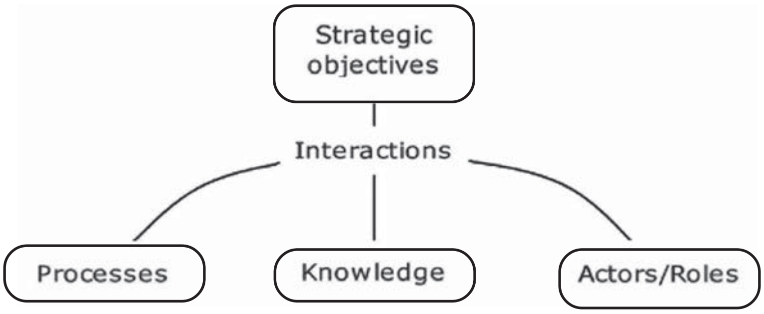

Figure 1. ***Goals of MIMIK***

creation cycle. The goals of MIMIK are to identify (Fig. 1):

– Strategic goals of an organization
– Actors and roles
– Knowledge resources
– Processes
– Interactions between these elements

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 (Ross 1997). They can be found in all sectors of activity and do not have to be linked to an information system. Some of them are implicit, meaning that they are not written anywhere but they belong nevertheless to the "business culture". However the basic formalism proposed by Ross (1997) is not sufficient in all cases to model the "know-why". Indeed, a "knowledge unit" 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 (Fraser & al. 2003) and business rules cannot model all of these types of knowledge. A different approach is described by authors such as Gamper & 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 datamodel for e-Government (Glassey 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 modelling.

However ontologies still cannot represent complex knowledge such as storytelling or human advice. As Samuel Johnson put it in the 18th century already: "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" and to describe this knowledge.

We identified several attributes for knowledge components. Capurro (2004) compares the knowledge typology proposed by Zahn &

al. (2000) with the classical Aristotelian one. 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 digitalized), as opposed to implicit knowledge that can not 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 here we used these terms here in a limited sense, as defined in computer science:

- *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. 2 shows the complete component-based architecture under the form of a class diagram, but it can also be formalized in RDF.

We furthermore based our work on the model theory approach developed by Wyssusek & al. (2001) to integrate process modelling and

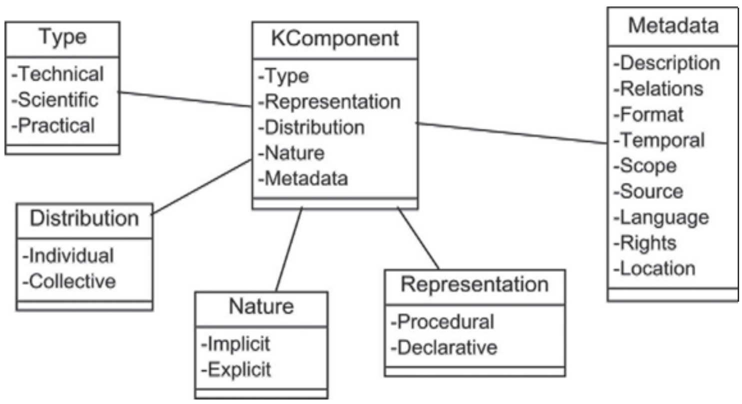

Figure 2. ***Knowledge component***

knowledge management. They provide an epistemological foundation to justify their approach, but they do not offer any practical methodology or examples. That is why we created a conceptual framework that aimed at the integration of both these approaches.

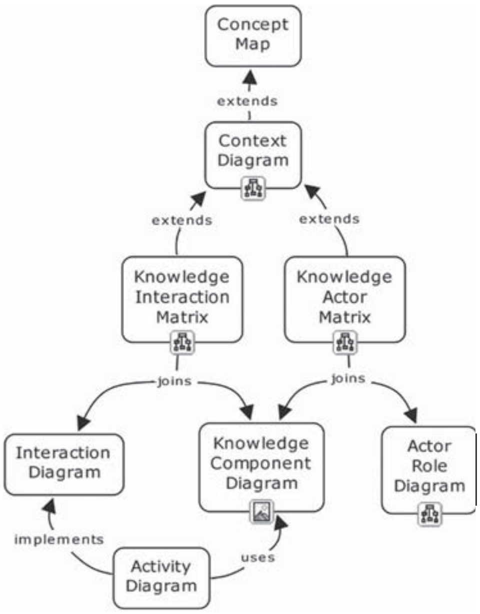

Figure 3. ***MIMIK Metamodel***

MIMIK consists of 8 types of diagrams (Fig. 3), most of them being inspired or directly taken from existing modelling techniques, mostly UML. As in UML or other modelling 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 modelling. We will provide examples for most of these diagrams, but more explanations can be found in Glassey (2005).

Concept maps (*Cmap*) are the top-level diagrams and show the strategic goals of an organization in terms of functions or processes: the metamodel (Fig. 3) of our framework is in itself a concept map. These concept maps can be decomposed in several levels, a terminal node of this type of diagram is implemented by a context diagram (*Cdiag*).

Context diagrams (Fig. 4) are almost exactly the same as use cases in UML, but we added the concept of knowledge packet. A knowledge packet is an abstract representation of a set of knowledge components (*Kcomp*, such as described in Fig. 2). These components encapsulate documents, databases, files, implicit knowledge and so on. They provide metadescriptions for "knowledge units", thus allowing us to show what type of knowledge is necessary in order to complete a process and which knowledge is relevant in a given context. Context diagrams provide an abstract view of the "know-what".

To model organizational structure we used actor-role diagrams, which allowed us to established formal distinction between actors and roles. These actor-role diagrams can be either classical organizational charts or matrices that formally link actors and roles in cases where the organization is too complex to be shown graphically in an intelligible way. We will not show that here, but let use mention that the abstract actors represented in context

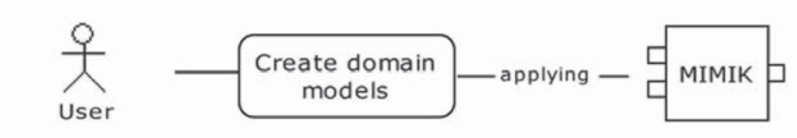

Figure 4. ***Context Diagrams***

diagrams can be linked to these actor-role diagrams. Moreover the actors described in such diagrams are used in the knowledge-actor matrices (see further on).

Fig. 5 shows a knowledge-interaction matrix, formally linking knowledge components to the interactions that implement a use case. 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 this matrix. By matching the "know-why" (knowledge components) and the "know-how" (interaction diagrams), this matrix shows the "know-what" at the operational level.

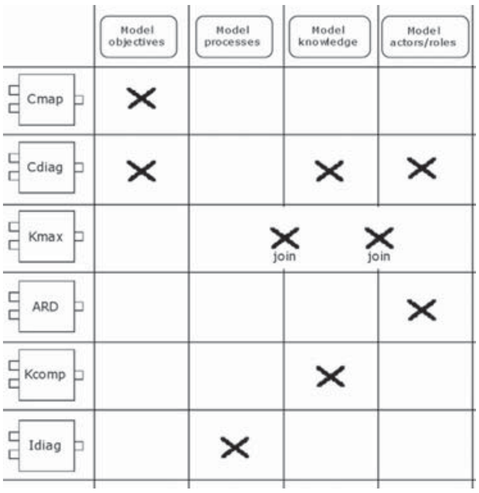

Figure 5. ***Knowledge-Interaction Matrix***

Exactly as knowledge-interaction matrices link knowledge components and interactions, the concept of knowledge-actor matrices (Fig. 6) create a formal relation between knowledge components and real actors within an organization. They provide an organizational view of the "know-what" or more precisely they show the "who-knows-what". That proves very useful in order to introduce implicit knowledge in a graphical model: it might not be possible to transform it into explicit knowledge but at least we know who has this knowledge within an organization. Knowledge-interaction matrices can also link actors and interaction diagrams provided a small constraint: within interaction diagrams modellers should only use roles that were defined in actor-role diagrams.

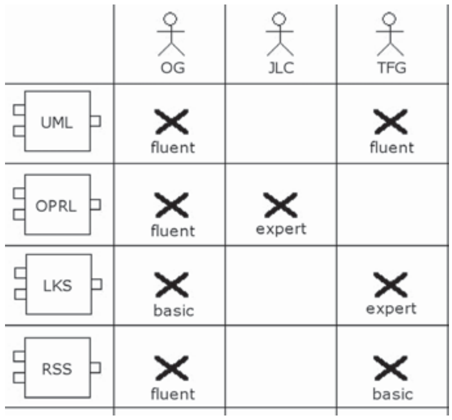

Figure 6. ***Knowledge-Actor Matrix***

The concept of knowledge matrix (*Kmax*) provides a formal link between the strategic and operational models:

- Terminal nodes of a concept map are linked to context diagrams
- Context diagrams describe processes in context, by linking them to knowledge packets and actors (strategic view)
- Knowledge matrices formalize the interactions between processes and knowledge (knowledge-interaction matrix) and processes and actors (knowledge-actor matrix) at the operational level.

At the operational level MIMIK uses standard UML collaboration and sequence diagrams, grouped under the name of interaction diagrams (*Idiag*), a given process in a knowledge-interaction matrix being formalized by one or several interaction diagram(s). The very detailed level is provided by UML activity diagrams. Knowledge components (*Kcomp*) are integrated as objects (or instances of the Kcomp class) in interaction and activity diagrams.

Prototype

The prototype of a knowledge-sharing system (*KnowS*) is based on RSS (Really Simple Syndication). RSS is a family of XML file formats for web syndication. 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 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" should validate this new content before it is available to anyone, but it is not required. In our Herford example, clerks would be notified when colleagues have found new pieces of knowledge in legal databases, online law commentaries, or when they have themselves implemented a new form or a new calculation formula. A moderator could then validate or complete this knowledge published by a clerk. All clerks could then rely on this knowledge in their daily work, as it would have been validated by a "domain expert". 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 advances 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 extend, email clients offer powerful content management capabilities, such as filters to limit access to only relevant content: a user can for example subscribe only to feeds that aggregate content on social welfare issues and limit this to parent support, they can furthermore implement filters stating that all feed elements not concerning their own domain. This is very useful to avoid information overflow, that is to limit the risk that the users will not read the feeds anymore because they receive to much irrelevant information.

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.

We built the KnowS prototype (Fig. 7) on a dedicated blogging platform: WordPress offers tools for categories' management and 10 levels of users' rights (read, write, edit, validate, publish...). By using categories for blog entries, we could 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. Let us also mention that the KnowS prototype integrates 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

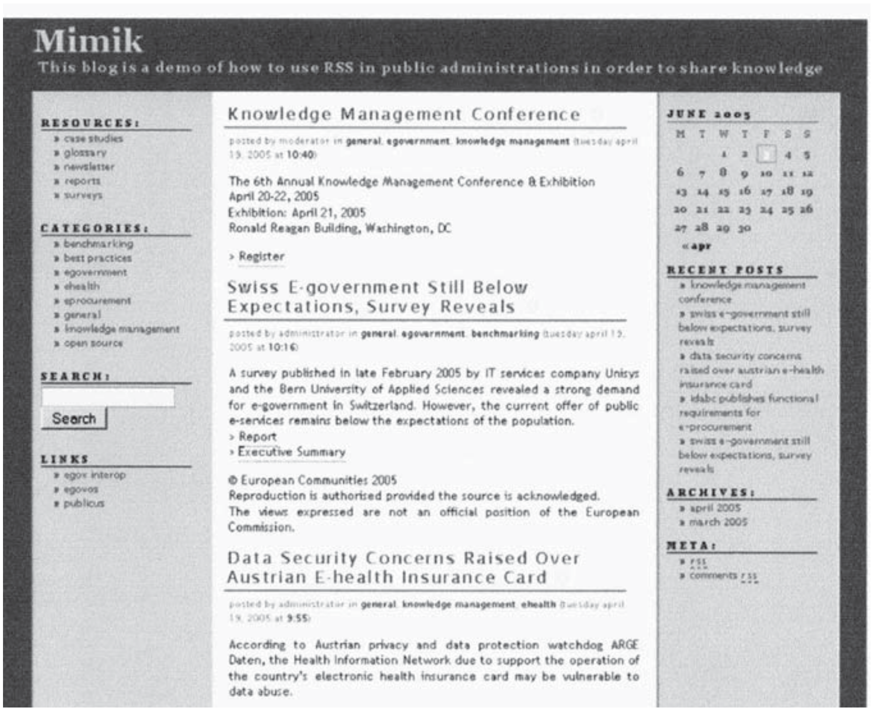

Figure 7. ***Web interface to the MIMIK demo blog***

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).

Conclusions

As we mentioned in the introduction, the MIMIK framework is focused on knowledge visualization and in this paper we showed several types of visualization:

- Concept maps can be used to create heuristic sketches;
- Context diagrams, interaction diagrams and knowledge components diagrams are typical conceptual diagrams;
- Knowledge matrices provide basic knowledge maps.

In the introduction we also defined fundamental concepts of knowledge creation and transfer: Fig. 8 provides a synthetic view of the knowledge creation life-cycle.

The MIMIK framework supports this knowledge creation life-cycle:

- Modelling high-level domain knowledge with concept maps and context diagrams facilitates the process of representing abstract knowledge in a more visual and explicit form (externalization).
- Knowledge matrices are efficient tools to organize knowledge.
- Knowledge components and RDF/XML are used to formalize knowledge.
- Knowledge components and RSS/XML technologies provide a platform for combination and distribution of knowledge
- A knowledge-sharing system (such as the *KnowS* prototype) makes it possible to share knowledge within and across organizations, and its users will have a tool that facilitates the interiorization of relevant knowledge.

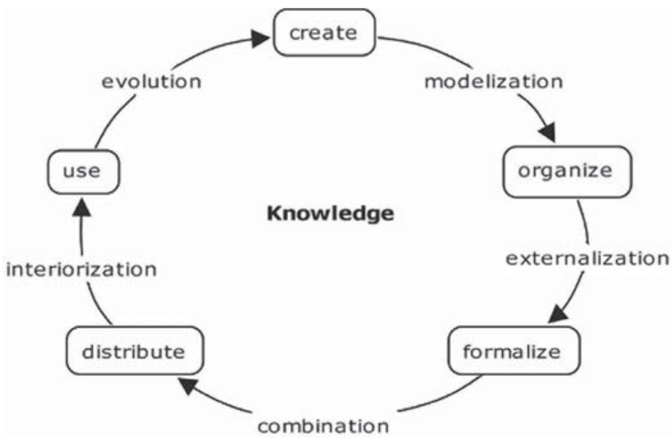

Figure 8. ***Knowledge creation cycle and MIMIK models***

REFERENCES

ALAVI, M. & LEIDNER D. E. (2001). Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS Quarterly*, vol. 25(1), p. 107–136.

CAPURRO, R. (2004). Skeptical Knowledge Management. In H. C. Hobohm (Ed.). *Knowledge Management: Libraries and Librarians Taking up the Challenge.* Munich: Saur, p. 47–57.

EPPLER, M. J. & BURKHARD, R. A. (2004). *Knowledge Visualization*. Retrieved 21.04.2005, from http://www.knowledgemedia.org/modules/pub/view.php/knowledgemedia-67.

FRASER, J.; ADAMS, N.; MACINTOSH, A.; McKAY-HUBBARD, A.; PARIENTE, Lobo T. & FERNANDEZ PARDO P., et al. (2003,). Knowledge Management Applied to E-government Services: The Use of an Ontology. In *Proceedings 4th International Working Conference KMGov 2003*. Heidelberg: Springer Verlag.

GAMPER, J.; NEJDL, W. & WOLPERS, M. (1999). Combining Ontologies and Terminologies in Information Systems. Paper presented at the *5th International Congress on Terminology and Knowledge Engineering*, Innsbruck, Austria.

GLASSEY, O. (2004). Developing a one-stop government data model. *Government Information Quarterly*, vol. 21(2), p. 156–169.

GLASSEY, O. (2005). Knowledge Component-based Architecture for Process Modelling. Paper presented at the eGOV INTEROP'05 Conference Geneva, Switzerland, February 23–24, 2005.

GORDON, T. (2003). An Open, Scalable and Distributed Platform for Public Discourse, Informatik 2003. *Informatik*, vol. 2, p. 232–234.

GRUBER, T. (1993). Towards principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies,* vol. 43(5–6), p. 907–928.

McLURE, Wasko, M & FARAJ, S. (2005). Why should I share? Examining Social Capital and Knowledge Contribution in Electronic Networks of Practice. *MIS Quarterly,* vol. 29(1), p. 35–57.

NISSEN, M. E.; KAMEL, M. N. & SENGUPTA, K. C. (2000). A Framework for Integrating Knowledge Process and System Design. *Information Strategy: The Executive's Journal*, vol. 16(4), p. 17–26.

NONAKA, I. & TAKEUCHI, H. (1995). *The Knowledge Creating Company: How Japanese Companies Create the Dynamics of Evolution*. Oxford University Press.

NOY, F.; FERGERSON, R. W. & MUSEN, M. (2000). The Knowledge Model of Protégé-2000: combining interoperability and flexibility. Retrieved 12.01.2003, from http://www.smi.stanford.edu/projects/protege/

RAINIE, L. (2005). The State of Blogging, Pew Internet & American Life Project, February 2, 2005. Retrieved 10.03.2005, from http://www.pewinternet.org/

ROSS, R. (1997). *The Business Rule Book: Classifying, Defining and Modelling Rules, Version 4.0*. Houston, TX: Business Rules Solutions.

SCHEER, A. W. (2001). *ARIS – Modellierungsmethoden, Metamodelle, Anwendungen*. Berlin; Heidelberg: Springer-Verlag.

ULLMAN, M. T. (2001). A Neurocognitive Perspective on Language: The Declarative/Procedural Model. *Nature Reviews – Neuroscience*, vol. 2(10), p. 717–726.

WINER, D. (2005). RSS 2.0 Specification, RSS at Harvard Law. Retrieved March 10, 2005, from http://blogs.law.harvard.edu/tech/rss

ZAHN, E.; FOSCHIANI, S. & TILEBEIN, M. (2000). Nachhaltige Wettbewerbsvorteile durch Wissensmanagement. In *Wettbewerbsvorteile durch Wissensmanagement*: Methodik und Andwendungen des Knowledge Management. Stuttgart, p. 239–270.

NUO GRAFINIO MODELIAVIMO LINK TARPORGANIZACINIO ŽINIŲ DALIJIMOSI VIEŠAJAME SEKTORIUJE

Olivier Glassey

Santrauka

Straipsnyje analizuojama žinių modeliavimo ir dalijimosi viešųjų įstaigų viduje ir tarp organizacijų struktūrinė sandara bei žinių dalijimosi sistemos modelis. Pirmiausia pateikiamas trumpas teorinis žinių kūrimo, perdavimo ir pritaikymo procesų analizės pagrindas, supažindinama su paties autoriaus siūlomu žinių modeliavimo ir dalijimosi modeliu MIMIK (Integruotų žinių modeliavimo metodas ir instrumentai, angl. *Method and Instruments to Model Integrated Knowledge*). Lygindamas su esamais žinių kūrimo, perdavimo ir pritaikymo procesų modeliais, autorius argumentuotai pagrindžia savo siūlomo modelio pranašumus. Rezultatai pateikiami architektūrine modelio išklotine.

Įteikta 2006 m. kovo 9 d.