Collaborative enterprise integrated modelling[*]

Chiara Ghidini[1], Marco Rospocher[1], Luciano Serafini[1], Andreas Faatz[2], Barbara Kump[3], Tobias Ley[4], Viktoria Pammer[3], and Stefanie Lindstaedt[4]

[1] FBK-irst, Via Sommarive 18 Povo, 38050,Trento, Italy
[2] SAP Research, SAP AG. Bleichstraße 8, 64283 Darmstadt, Germany
[3] Knowledge Management Institute, TU Graz. Inffeldgasse 21a, 8010 Graz, Austria
[4] Know-Center, Inffeldgasse 21a, 8010 Graz, Austria.

Abstract. Enterprise modelling focuses on the construction of a structured description, the so-called *enterprise model*, which represents (a subset of) the aspects relevant to the activity of an enterprise. Nowadays, knowledge engineering provides sophisticated methodologies and tools to support enterprise modelling and production of formal enterprise models. Recently, it has become clearer that enterprise modelling is a collaborative activity; however, most of the modelling techniques and tools devised so far do not consider this collaborative dimension properly. Therefore, we envisage the necessity to expand, extend, and generalize current methodologies in order to exploit this new dimension. In this line, we propose a flexible framework for enterprise modelling that supports an agile collaboration between the actors involved in the modelling activities, without sticking to a pre-defined protocol of interaction. This methodology produces an integrated enterprise model model that, not only contains a structural (formal) description, but it also includes more informal and semi-formal information. We also present a tool, based on a semantic wiki, that supports the proposed methodology.

1 Introduction and motivations

An enterprise model [1] is a description in structured terms (i.e., objects and relations) of one or more aspects of an enterprise. An *integrated* enterprise model is obtained by combining a set of enterprise models, each of which describes a specific aspect of the enterprise[1]. The combination is obtained by making explicit the relations between the different component models as intra-model links. *Enterprise modelling* is the process of creation of an (integrated) enterprise model. Often, an integrated enterprise model focuses in the description of two specific aspects of an enterprise: (i) its processes and activities, and / or (ii) the business domain within which the enterprise operates. Recently, other aspects of an enterprise, like goals, human resources, the enterprise structure and roles, and so on have become important assets to be described in an enterprise model. This is due to the central role that enterprise models are taking in the development of a large number of applications, including Internet and (Semantic) Web based applications.

[*] This work has been partially funded under grant 027023 in the IST work programme of the European Community (APOSDLE IST-project).

[1] In the rest of the paper, for readability reasons, we sometime omit the word “integrated”.

Building an enterprise model requires a number of skills. These skills span from *knowing* the different aspects that have to be described in the models, to having the ability of *encoding* such knowledge into formal statements, to having the ability of *integrating* different aspects (models) into a uniform and coherent vision. For this reason, enterprise modelling is inherently a collaborative activity, performed by different actors and carried on under some collaborative protocol, which is usually described in the methodology used to support the modelling.

Current modelling methodologies [2] are based on a specific collaborative protocol: the, so-called, *waterfall* paradigm. In this paradigm the starting point in modelling is a collection of informal knowledge provided by knowledge experts; this knowledge is transformed by knowledge engineers into a set of formal statements (possibly with the support of semi-automatic transformation tools), which constitute the final model. Intermediate evaluation steps are possible only at certain stages of the process. An approach based on the waterfall paradigm presents two orders of problems. First, the collaboration pattern has to stick to rigid interaction protocols: for instance, the knowledge experts cannot insert new knowledge or modify already specified knowledge at any time during the modelling process. Second, the waterfall paradigm does not support a reverse transformation from formal knowledge to informal knowledge. This transformation would be extremely useful to support a real collaborative modelling between knowledge experts and knowledge engineers. For instance, the knowledge engineer could use this to validate the encoding of informal knowledge into a formal model, and, vice-versa, the knowledge expert could use this to monitor the evolution of the formal model.

A further limitation of current modelling methodologies is that, after a limited initial phase of informal specification, they tend to modularise and separate the transformation of informal knowledge into the different formal models covering the different aspects, with very few check points for integration. The drawback of this approach is the risk of generating unaligned models, which are difficult to integrate in a coherent formal framework. In fact, modelling methodologies are mainly focused towards the representation of single aspects of enterprises, with some exception focusing on the integration of domain with processes. One reason for this is the lack of reference meta-models to be used in the production of integrated enterprise models.

The contribution of this paper is to support the collaborative modelling activity in two different forms: first, we propose a flexible modelling workflow, where different modelling actors can asynchronously collaborate without sticking to a pre-defined modelling workflow; second, we propose an integrated meta-model that supports the integration of single components of an enterprise as well as their intra-model relations. This model was devised to support the development of work integrated learning applications and integrates a domain specific model, a process model, and a competency model.

2 A collaborative framework

Enterprise modelling is a process which involves a number of actors, is organised according to a modelling protocol, and produces a result which is the enterprise model. In the following we provide an overview of our approach towards enterprise modelling, by describing how we represent these three aspects: (a) the actors, (b) the modelling protocol, and (c) the result.

Actors. Enterprise modelling requires actors (roles) with different, and often complementary, knowledge and skills. Two skills that are typically involved in modelling an enterprise are:

- a deep understanding of the domain, or aspect of the domain, to be modelled;
- the capability of translating informal knowledge on a domain into adequate mathematical/computational models.

The actors typically involved in enterprise modelling are usually specialised in only one of these two skills and are usually referred to as:

Knowledge expert. This actor owns knowledge about a domain, sometimes implicitly. (S)He usually does not have any skill in making this knowledge explicit in a formal model;

Knowledge engineer. This actor has the capability of encoding a piece of informal knowledge into a formal model. Usually (s)he has no, or very little, understanding of the domain to be modeled.

To better support the complex task of enterprise modelling, we envisage the need of a third actor (role), which we call *modelling facilitator*. The modelling facilitator sits between the knowledge expert and the knowledge engineer, and her/his main responsability is in designing, managing, and monitoring the entire process of enterprise model creation. Indeed the methodology and the tools we are proposing are likely to involve a number of steps and activities that should be tailored to the specific enterprise and domains to be modelled in order to optimise the elicitation process.

Protocol. The two basic steps described in state of the art methodologies for knowledge engineering are:

Knowledge elicitation. This consists either in the specification of portions of knowledge in some informal language (natural language or intermediate structured language), or in feedback on the formalisation and update of the informal specification accordingly;

Knowledge formalisation. This consists in the translation of the informal specification into a set of statements in a logical language, possibly maintaining some connection between the logical statements and the informal description they formalise.

Modelling methodologies provides detailed protocols that specify how and when each single activity should be carried out. We propose a agile approach in which the two activities are not carried on following a *synchronous* protocol. Instead, they are executed in parallel and they can interfere one with another, so that the following requirements are met:

- the knowledge expert shall be able to monitor the entire modelling process not only by checking the final result, but also inspecting intermediate steps. For example (s)he can comment on a portion of formal knowledge while it is created, without waiting until the entire model is constructed;
- the knowledge engineer shall be able to ask the knowledge expert to provide "real time" and agile feedback on new constraint (e.g. simplifications and abstractions) introduced during the encoding of informal knowledge into formal statements.
- an automatic transformation of the results of the activity of the knowledge expert in the formal language of the knowledge engineer (and vice-versa) shall be provided.

To support the protocol we envisage we propose a *collaborative enterprise modelling* paradigm, illustrated in Figure 1. This paradigm is inspired by recent Web 2.0 collaborative solutions, e.g. wikis. In this paradigm all the actors asynchronously collaborate towards the construction of an *integrated enterprise model* by inserting knowledge (either formal or informal), by transforming knowledge (from informal to formal) and by revising knowledge. The three boxes on top represent the three different types of actors involved in modelling. The knowledge experts enter the missing knowledge - in form of informal knowledge - into the models or provides feedback on the models. The result of this input is stored in the "informal part" of the model. The system semi-automatically translates part of the informal model into a formal specification and vice-versa. Asynchronously, the knowledge engineers can refine the "formal part" of the model by inserting new statements and adding new constraints. The modelling facilitators act as coaches, they monitor the entire process and, for instance, they can suggest part of the model to be specified and revised. They are also responsible for designing schematic templates to collect semi-structured knowledge, helping with the elicitation of knowledge from domain experts, and so on.

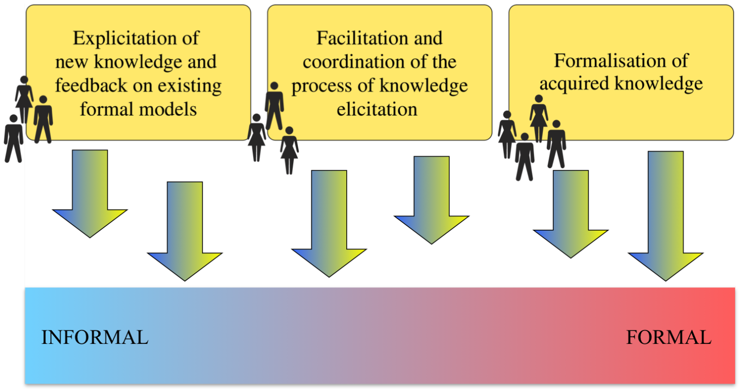
Explicitation of new knowledge and feedback on existing formal models
Facilitation and coordination of the process of knowledge elicitation
Formalisation of acquired knowledge
INFORMAL
FORMAL

Fig. 1. Collaborative enterprise modelling.

Result. The main goal of enterprise modelling is traditionally the production of an integrated formal model in which the different aspects of an enterprise are integrated in a unique model. Differently from what happens in software engineering, where the result is a “program” which is executable by a computer, in knowledge engineering, the result of modelling is an artefact that can be used both by humans and machines. Indeed, enterprise models are created to support the implementation of technological tools as well as the organisation of people activities. This means that formal models describe knowledge that can be exploited by computers but also by people who, in most of the cases, do not have competencies in understanding formal languages and representations. To support the exploitation of an enterprise model also by humans we require that the result of enterprise modelling is an artefact in which the informal knowledge is also integrated with the formal models. An example of the type of integration we thinking of can be found in [3], in which each formal statement is aligned with a paragraph of natural language that informally describes the statement. In our work we want to pursue this direction, by supporting a tight integration of formal and informal knowledge.

3 The enterprise meta-model

The structure of the enterprise model (also referred as *the meta-model*) contains informal, semi-formal, and formal information about key aspects of an enterprise. In this paper we focus on the business domain, the processes, and the competencies of an enterprise[2]. All the informal and semi-formal portion of these three sub-models are represented by means of a semantic wiki, while the formal part of each sub-model is represented by an appropriate formalism.

The domain model provides the description of the business domain within which the enterprise operates. It is a conceptualisation of the entities and the relations between them, which are produced, consumed, or are simply relevant to the activities of an enterprise. This description is provided in terms of *concepts*, *relations*, and *objects*.

Conceptual Modelling and Knowledge Representation provide sets of mature of-the-shelf approaches and tools for the formal representation of a specific domain. Following the growing popularity of Semantic Web technologies, we decided to base our representation of a domain-specific model upon the OWL ontology language[3]. This approach allows one to express classes, properties, instances, and axioms among them. We build on top of these results, focusing more on the problem of how to support the construction of a domain-specific ontology, embedded in a integrated enterprise model, rather than developing new representation paradigms. The representation of the domain-specific model as an OWL ontology can be used inside the integrated enterprise model, in applications that require a comprehensive and coherent overview of multiple aspects of the enterprise, and as a stand-alone ontology, in typical applications of Semantic Web technology such as information integration and reasoning, extraction of knowledge from text, annotation and tagging, and so on.

[2] More complex enterprise models can be considered, and we have designed our approach with the explicit intent to be open and extendable to other aspects of an enterprise

[3] www.w3.org/TR/owlfeatures/

The process model provides a description of the patterns and procedures occurring in a business domain of an organisation. The very core of a process model is a control flow.

A number of different approaches, formalisms and techniques exist to model processes. They depend upon several factors. Among them a crucial one is the degree of flexibility and continuous variation of the nature and order of the steps performed during the daily work. While more stable routine work (e.g. travel expense accounting) is usually formalised by means of workflows [4] and formal models based on Petri Nets [5], more flexible and dynamic process models have been proposed from experiences in agile software development (a typical example of flexible process model is *scrum* [6]). In practice, depending on the application it is possible different formalisms can be adopted, depending from the complexity of the model. For instance, in the e-learning application scenario described in [7] it was enough to consider task to be either atomic, or composed of a bag of subtasks, regardless from the execution control. In this case a simple hierarchical structure representing the task/sub-task relation was sufficient and we adopted an OWL ontology that encodes the part-of relation. In the situation in which tasks are complex structures described, for instance, by BPMN[4] language [8], we adopt a more complex model, in which processes are described by means of the primitives defined in an OWL ontology that represents BPMN[5].

The competency model describes the attitudes and the capability of people employed in an organisation to fulfill their tasks and to reach their objectives and goals. Competency models can be build with many goals in mind (such as, improving enterprise training, selection and placement, knowledge management, expertise recommendations, team building and many others): in this paper, we focus on the support of individual learning in the process of working tasks [9].

The competency model describes users in terms of knowledge about concepts of the business domain and skills to perform the tasks of the process model. Because skills are assigned to tasks, as well as to domain model elements, the competency model constitutes the connection between the domain model and the process model. Practically, this connection is established by assigning tasks to competencies (domain model element plus skill type) which are required for performing the task. By applying Knowledge Space Theory, we derive a partial order on sets of skills, which formalises a prerequisite relation. This can be used for defining the teaching model of an adaptive learning environment (e.g. making inferences about optimal sequencing for learning) or updating the user model (e.g. adaptive competency diagnosis)[10].

4 Enterprise modelling using the MoKi

The MoKi (**Mo**delling Wi**ki**) is a collaborative tool that (i) supports access to the enterprise model at different levels of formality (informal, semi-formal and formal), (ii) supports integrated modelling of domain, processes and competences within an enterprise, and (iii) ensures a coherent development of the formal part.

[4] Business Process Modelling Notation `www.bpmn.org`

[5] `http://dkm.fbk.eu/index.php/BPMN_Ontology`

The proposed collaborative tool is based on SMW[6] and integrates different views over portions of knowledge. The basic idea is to create a SMW page for each element of the formal model. The content of the page is not only an informal description of the element (e.g. a concept, a relation, a task, and so on) but also a semi-formal and a formal representation of the properties of the element.

Description This part contains an informal description of the element considered, mainly in the form of natural language sentences. Images or drawings can be used as well. This is like a usual wiki page. No specific syntax, or modelling expertise is required to edit this section. The basic editing functionalities are provided by the MediaWiki toolbar.

This portion might contain also a part named *Open issues/feedback* where the knowledge expert can ask for, and be provided with, feedback on the models created by means of yes/no answers to some precise and unambiguous statements. These statements are proposed to the knowledge expert by the modelling facilitator, which can create them by hand or with the help of some tools for the automatic generation of natural language statements from the formal models (e.g. [11]).

Semi-formal representation In this part, the element is described by means of templates, that can be filled by the knowledge expert. The templates are designed (or selected from a library) by the modelling facilitator. It is always possible for the knowledge expert to insert semi-formal information by means of open templates, i.e. templates which do not respect any predefined structure. This can be a table, a graph, etc. The clear advantage of describing an element this way is that the content of templates can be automatically translated to a formal language (e.g. OWL).

Formal representation In this part, the element considered is described according to the formal language chosen. To support models editing/browsing facilities, state of the art tools for formal model creations (e.g. Protégé for the domain ontology or a BPMN editor for processes) can be directly invoked from within the section. Changes to the formal models are then propagated to the OWL integrated models. The main actor involved in maintaining this section of the page is the knowledge engineer.

Knowledge experts, modelling facilitators and knowledge engineers can also interact with each others and exchange further ideas and comments using the *discussion* SMW's built-it functionality. To support modelling the MoKi provides groups of functionalities, described below, that can be accessed via a wiki's style menu bar. Note that in the description below we use the term model element to indicate the basic components of the model. For instance, a concept or a relation of the domain model, a task of the process model, a competency, and so on.

Import Functionalities These functionalities, which can be used as basis for constructing model components, support import of formal models and their translation in informal/semi-formal format, easy input of lists of concepts (tasks) possibly organized according to predefined semantic structures (e.g. a taxonomy or a mereology), and automatic extraction of relevant terms from digital resources.

[6] Semantic Media Wiki `http://semantic-mediawiki.org`

Model Management This is a set of functionalities that support the knowledge experts and the knowledge engineer in the creation, editing and deletion of pages that describe model elements. Actors may have restrictive access to different parts of the page depending on their roles. For instance, a knowledge engineer is allowed to access only the formal part of the page, while a knowledge expert can access only the informal or semi-formal part.

Visualization These functionalities allows to produce different types of graphical overviews of the models. These functionalities help the actors to deal with the global picture and not only with model elements.

Export functionalities These functionalities support the automatic export of knowledge of the enterprise model into a standard knowledge representation languages, such as OWL, BPMN, YAWL, and so on.

The current version of the MoKi[7] supports all the above functionalities with the exception of the insertion of formal statements, the editing of competency model, and the management of access permissions. Also, concerning the informa/semi-formal/formal structuring of a page we currently focus on the semi-formal part, which also contains some informal description. The formal part is not currently included in the MoKi but can be automatically obtained using the export functionality. A detailed description of the current version of the MoKi is contained in the MoKi manual [12].

The current version of the MoKi is currently being used by 4 Application Partners inside the APOSDLE EU-project[8] to produce enterprise models for their specific scenarios. An in-depth evaluation of the MoKi tool is planned starting from the data that will be available at the end of the modelling process made by the knowledge experts of the 4 different use cases.

5 Related Work

Solutions to the problem of modelling various aspects of an enterprise were proposed in several works, both in terms of definition of the meta-model and in terms of methodologies to support the creation of the model itself. Concerning the meta-model, the TOVE Ontology Project [1] proposes a set of integrated ontologies for the modelling of an enterprise which spans several aspects of an enterprise, such as activities, states, resources, time, and so on. The Enterprise Ontology [13] proposes to use five top-level classes for integrating the various aspects of an enterprise: meta-ontology, activities and processes, organisation, strategy, and marketing. The Common KADS model set [14] is a collection of models (organisation, task, agent,...) for structuring knowledge in an organisation. The novelty of our approach w.r.t. these proposals lies in integrating formal and informal knowledge, and also to include competencies within the general schema. More in detail we we extend the approach to enterprise model proposed in [15] by introducing the collaborative dimension in the modeling of process and competencies.

Focusing on methodologies for ontology / model creation we can notice that most of them - e.g. TOVE Enterprise methodology [1], CommonKADS [14], Uschold and

[7] See `moki.fbk.eu`

[8] www.aposdle.org

King [16], and Methontology [17] - are built around the knowledge engineer, who executes and coordinates all the different phases of the knowledge acquisition and formalisation process. Few methodologies, such as DILIGENT [18], support collaborative ontology engineering, but they do not address the modelling of processes, competencies, and their integration with a domain ontology. The methodology which fits best to the idea of a collaborative framework presented in Section 2 is the recent NeOn methodology [19], which provides an explicit attempt to support the collaborative aspects of ontology building and also to support ontology practitioners rather than knowledge engineers.

Many tools are available to support the creation of formal models. Most of them, e.g. Protégé (protege.stanford.edu), were born as standalone desktop applications. Despite the development of pug-ins that support collaborative features (e.g., Collaborative Protégé) the tools remain barely usable by users with limited expertise of formal languages. Recently, wiki systems, and semantic wikis, have been applied to support collaborative knowledge creation and sharing. Among the relevant ones are: Semantic MediaWiki, IkeWiki (ikewiki.salzburg-research.at), Makna (makna.ag-nbi.de) and AceWiki (attempto.ifi.unizh.ch/acewiki). The recently started KiWi EU-project (www.kiwi-project.eu) has the main objective to develop an advanced knowledge management system based on a semantic wiki. However, all these approaches do not support the specification of processes and competencies, neither an automatic bi-directional transformation of informal specifications to formal models.

6 Conclusions

In this paper we have presented a new collaborative framework and an integrated meta-model for the modelling of an enterprise model. This work constitutes a genuine contribution towards supporting a fruitful collaboration among people with different skills and levels of expertise in the modelling activities. Moreover, the enterprise meta-model, initially devised for work-integrated learning applications, provides a general schema for the integration of formal and informal knowledge over fundamental aspects of an enterprise, that is, domain of affairs, processes, and competencies. The implementation of a tool for collaborative modelling inside the APOSDLE EU-project[9] already provides key functionalities towards the modelling of an integrated enterprise model in a collaborative manner, and constitutes a first version towards the realisation of the full framework.

References

1. Fox, M.S., Grüninger, M.: Enterprise modeling. AI Magazine **19**(3) (1998) 109–121
2. Cristani, M., Cuel, R.: A survey on ontology creation methodologies. Int. J. Semantic Web Inf. Syst. **1**(2) (2005) 49–69
3. Seyfang, A., Martínez-Salvador, B., Serban, R., Wittenberg, J., Miksch, S., Marcos, M., ten Teije, A., Rosenbrand, K.: Maintaining formal models of living guidelines efficiently. In:

[9] www.aposdle.org

11th Conference on Artificial Intelligence in Medicine. Volume 4594 of LNCS., Springer (2007) 441–445

4. Georgakopoulos, D., Hornick, M., Sheth, A.: An overview of workflow management: From process modeling to workflow automation infrastructure. Distributed and Parallel Databases **3**(2) (1995) 119–153
5. van der Aalst, W.M.P.: The application of petri nets to workflow management. Journal of Circuits, Systems, and Computers **8**(1) (1998) 21–66
6. Rising, L., Janoff, N., Syst, A., Phoenix, A.: The Scrum software development process for small teams. Software, IEEE **17**(4) (2000) 26–32
7. Christl, C., Ghidini, C., Guss, J., Lindstaedt, S.N., Pammer, V., Rospocher, M., Scheir, P., Serafini, L.: Deploying semantic web technologies for work integrated learning in industry - a comparison: Sme vs. large sized company. In: International Semantic Web Conference. Volume 5318 of Lecture Notes in Computer Science., Springer (2008) 709–722
8. Francescomarino, C.D., Ghidini, C., Rospocher, M., Serafini, L., Tonella, P.: Reasoning on semantically annotated processes. In: Proceedings of the 6th International Conference on Service Oriented Computing (ICSOC'08), Sydney, Australia (2008)
9. Lindstaedt, S.N., Ley, T., Scheir, P., Ulbrich, A.: Applying Scruffy Methods to Enable Work-integrated Learning. Upgrade (2008) in press
10. Ley, T., Kump, B., Ulbrich, A., Scheir, P., Lindstaedt, S.N.: A competence-based approach for formalizing learning goals in work-integrated learning. In: Proceedings of Ed Media 2008 World Conference on Educational Multimedia, Hypermedia & Telecommunications, Vienna, Austria, June 30-July 4. AACE, Chesapeake, VA (2008) in press
11. Kaljurand, K., Fuchs, N.: Verbalizing OWL in Attempto Controlled English. In: OWLED 2007. (2007)
12. Ghidini, C.: How to use moki `https://aposdle.fbk.eu/AP1/`.
13. Uschold, M., King, M., Moralee, S., Zorgios, Y.: The enterprise ontology. www.aiai.ed.ac.uk/ entprise/enterprise/ontology.html (1995)
14. Schreiber, G., Akkermans, H., Anjewierden, A., Dehoog, R., Shadbolt, N., Vandevelde, W., Wielinga, B.: Knowledge Engineering and Management: The CommonKADS Methodology. The MIT Press (December 1999)
15. Gruninger, M., Fox, M.: Methodology for the design and evaluation of ontologies. In: IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal (1995)
16. Uschold, M., King, M.: Towards a methodology for building ontologies. In: Proc. of IJCAI95's WS on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada (1995)
17. Fernández-López, M., Gómez-Pérez, A., Juristo, N.: Methontology: from ontological art towards ontological engineering. In: Proc. Symp. on Ontological Eng. of AAAI. (1997)
18. Pinto, H., Tempich, C., Sure, Y.: DILIGENT: Towards a fine-grained methodology for Distributed, Loosely-controlled and evolvInG Engingeering of oNTologies. In: Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004), Valencia, Spain (August 22nd - 27th 2004)
19. Suarez-Figueroa, M.C., de Cea, G.A., Buil, C., Dellschaft, K., Fernandez-Lopez, M., Garcia, A., Gomez-Perez, A., Herrero, G., Montiel-Ponsoda, E., Sabou, M., Villazon-Terrazas, B., Yufei, Z.: D5.4.1. NeOn Methodology for Building Contextualized Ontology Networks. NeOn Deliverable. (February 2008)