CIRN Prato 2009 Keynote Address

Collaboration Patterns as Building Blocks for Community Informatics[1]

Aldo de Moor

CommunitySense, Tilburg, the Netherlands
ademoor@communitysense.nl

Abstract. Community Informatics is a wide-ranging field of inquiry and practice, with many paradigms, disciplines, and perspectives intersecting. Community informatics research and practice build on several methodological pillars: contexts/values, cases, process/methodology, and systems. Socio-technical patterns and pattern languages are the glue that help connect these pillars. Patterns define relatively stable solutions to recurring problems at the right level of abstraction, which means that they are concrete enough to be useful, while also sufficiently abstract to be reusable. The goal of this paper is to outline a practical approach to improve CI research and practice through collaboration patterns. This approach should help to strengthen the analysis, design, implementation, and evaluation of socio-technical community systems. The methodology is illustrated with examples from the ESSENCE (E-Science/Sensemaking/Climate Change) community.

Keywords: Community Informatics, research, practice, pattern languages, collaboration patterns, design, socio-technical systems

1 Introduction

Community Informatics (CI), with its related field of Development Informatics, is a wide ranging field of inquiry and practice. From its inception, CI referred to both a research area and the practice of enabling communities with ICTs (Gurstein 2008). These technologies should be conceived of as covering the complete spectrum from face-to-face meetings to sophisticated computer- and network-based tools. The communal needs that these technologies satisfy and the ways they are appropriated by their communities of use are not static. Instead, communities and their technologies co-evolve in that the technologies both afford and constrain the behaviour of their communities, and, in turn, these communities shape the technologies as they are being applied in messy practice (de Moor and Aakhus 2006). For example, Twitter allows people to widely distribute proto-ideas to a large audience. However, as the tool gets used more seriously, many people have expressed a need for better archiving functionalities for selected "tweets". Such forces result in complex, evolving socio-technical community systems. The ultimate user requirement that these systems should satisfy is legitimate interaction. This is to be achieved by translating social

[1] Community Informatics Research Network Conference, Prato, Italy, November 4-6, 2009

requirements like freedom, privacy, and democracy into technical specifications (Whitworth 2006).

CI research and practice are two sides of the same coin. Community based research is about "getting somewhere" (Stoecker 2008). Practice is what ultimately matters, leading to meaningful changes in the fabric of society. However, uninformed practice, especially in the CI domain that is so socially and politically sensitive and consisting of such fragile socio-technical constellations, leads to much wasted time and energy, or worse. An in-depth, systematic research approach, continuously informing and being informed by practice, is therefore of the essence in any real-world CI context.

Calls grow louder for CI research to focus more on the information and communication systems that increasingly circumscribe the way in which society - and the myriad communities which comprise it - operate and develop (Day and Schuler 2004; Gurstein and Horan 2005; Bieber et al. 2007; Stillman and Linger 2009). Many issues deserve critical attention – from global Internet infrastructure questions around net neutrality and the Digital Divide to practical design and implementation problems of a neighbourhood content management system. Unfortunately, there is still a large gap between CI research and practice and traditional IS research and development. This gap needs to be bridged as soon as possible in order to make the powerful mainstream information and communication systems that "run the world" much more sensitive to communal and societal needs. The main question is: how to link CI research to information and communication systems development methodologies? One way to answer this question is by making (still embryonic) CI research methodologies more systematically include systems development approaches. This would mean much more attention for the analysis, design, implementation, and evaluation processes of socio-technical community systems (Whitworth and De Moor, 2009).

CI is still a very young discipline. There is no such thing as *the* research methodology, nor will or should there ever be one size-fits all methodological straightjacket. The domain is simply too rich and pluriform and requires many different translations of core communal values such as mutuality, reciprocity, and respect (Williamson 2008). However, this does not mean that trying to find common ground in the way we frame our research questions, our approaches to address them, and the implementation of research results in real world contexts is without merit. On the contrary: especially when such fundamental communal and societal stakes are involved, it is our duty as practical researchers and researching practitioners to be very precise and try to find "unity in diversity" where we can. Without such common ground, critical questions are much harder to formulate, and lessons cannot be learnt and disseminated as efficiently. Rather than reinventing wheels over and over again, at great personal and societal cost, one should have a large but accessible catalog of wheels to choose from and be able to more easily customize these to the specific needs of a community of choice.

To find and this common ground, we first define four "methodological pillars" of CI research: contexts/values, cases, process/methodology, and systems. Next, we focus on socio-technical pattern languages and introduce the idea of collaboration patterns as the conceptual building blocks for CI research and practice. We then illustrate our approach with examples from the real-world case of the ESSENCE (E-

Science/Sensemaking/Climate Change) community. We end the paper with a discussion and conclusions.

2 Methodological Pillars of Community Informatics Research and Practice

Community Informatics is a "meta-field", which builds bridges across gaps between different social and technical paradigms as well as between theory and practice. In (de Moor 2009), we listed some practical steps to bridge these gaps, including becoming more precise about our definitions, more systematically identifying lessons learnt and best practices, and the establishment of testbeds and collaboratories to integrate theory and practice in socio-technical systems development. We also introduced four aspects of community informatics important for research: contexts/values, cases, process/methodology, and systems (Fig.1).

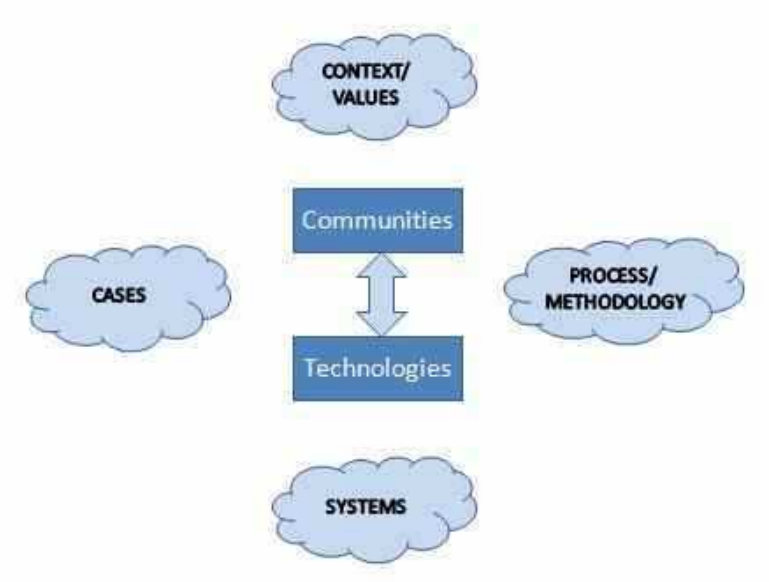

Fig. 1. Pillars of Community Informatics research

Here, we expand on these aspects and present them more clearly as methodological foundations of CI research. Although as stated above there is no clear CI paradigm yet, we contend that all of these pillars should at least be taken into account in any particular, more or less formal, CI research/practice methodology and the socio-technical systems that are both its development goal and object of evaluation and reflection.

2.1 Contexts/Values

Context is predominant in the CI tradition. There are two main types of contexts scoping our research and practice. All information, communication, and knowledge processes take place in some social context. The predominant type of CI context is rooted in community-centered development, focusing on fostering soci(et)al change, community empowerment, grassroots development, from the local all the way up to the global level. Examples of these contexts abound in, for instance, (Williamson and DeSouza 2007; Schuler 2008). Another main stream of CI research and practice revolves around knowledge sharing, which is more the focus of, for instance, Social and Library informatics, and much of the work done on online communities, Computer Supported Cooperative Work, and socio-technical systems design. Archetypical research objects here are communities of practice and interest found in all kinds of domains like research, education, business, knowledge management, and so on. Typologies of contexts there can be found in, for instance, (Preece 2000; Wenger 2002; Whitworth and Moor 2009). These are just two of the many possible ancestors of our young discipline, on which the jury is still out. The goal of this short introduction is not to be exhaustive, but exactly to show the many different contexts in which CI of some kind plays a role.

Independent of whether the community context is more of the social change or of the knowledge sharing type, they at least to some extent share a pool of common *values*. CI research, unlike more traditional branches of informatics, is very strong in analyzing the context of use of ICTs, including the stakeholders involved, their interests, norms, and goals, and cultures. In particular, much attention is paid to assessing the community values driving the development and uptake of these technologies. Whereas the design of traditional IS often focuses on efficiency, this may be less relevant in community contexts, where values like inclusiveness, representativeness, or equal access are often more important (Gurstein and Horan 2005). This focus on values is also exemplified by the human-centred approach to CI, including key notions like human purpose, cultural diversity, technology as tool, and social cohesion (Day 2005), or the core community values proposed by Schuler (2008, 62): conviviality and culture, lifelong education, strong democracy, health and well-being, economic equity, opportunity and sustainability, and open information and communication. Values can be used as motivation and criteria for change, must have precedence over the rules embodied in information and communication systems, and, as they differ per person and culture, need to resolve themselves in practice (Schuler 2008, 536, 22). If such context characteristics and values are not sufficiently taken into account, as is the case in many traditional IS development methodologies, the end result will often be information and communication systems that are “sterile” and do not agree with deeply felt communal or societal needs.

2.2 Cases

Community Informatics research is very much case-driven. Studying cases allows for obtaining an in-depth understanding (through explaining, exploring, and describing) of complex social phenomena, while retaining the holistic and meaningful characteristics of real-life events (Yin 1994). The human-centred design process that is the core of CI research requires that communities should be able to make their own qualitative, subjective judgements of a broad spectrum of considerations, which are different on a case-by-case basis (Day 2005). This results in rich, "lived" stories about how authentic information and communication requirements are being met in concrete communities, systems, and projects. In classic IS development methodologies,"user" requirement specifications tend to be much more abstract and separated from the unique characteristics of the community case at hand.

2.3 Process/Methodology

This is both a strength and a weakness of current CI research and practice. Our research and development tradition has created a large palette of situated communication, collaboration, development, evaluation and other community processes and methodologies. As a result, there is a very rich set of concepts, theoretical constructs, and theory overall in which to embed analysis and understanding (Gurstein 2008). Generally, the communities themselves play a more or less active role in these processes, from generating the research questions to playing other active roles in the research process (Stoecker 2008). Still, perhaps because these processes are so context- and case-dependent, lessons learnt are hard to generalize and reuse across cases. As a result, it is hard to go beyond "situated methodologies" and to build a solid methodological frameworks for more enlightened community systems development. Although valuable initial work has been done here (e.g. (Preece 2000), such methodologies are still in their infancy compared to the mainstream information systems development methodologies.

2.4 Systems

Much ICT research and development builds and evaluates individual tools in isolation, such as office applications, e-mail, and in the Web 2.0 age social software such as blogs, wikis, Twitter, and Facebook. However, realistic use cases ever more require the analysis of complex "tool systems" of multiple technologies being applied in concert in a specific collaborative or social context of use. This includes the numerous tools often being used simultaneously in e-learning, research, business, or campaigning contexts (de Moor 2007). In such a systems view, processes, technology, information, and people need to be co-designed into calibrated, integrated, and evolving socio-technical wholes.

As with its processes and methodologies, this systems view is often left implicit or expressed only informally in CI research and practice. This makes it difficult to disseminate and critically evaluate findings more widely, and to use them to inform practical systems design and engineering processes. The paradox is that, in contrast

with mainstream IS development approaches, CI research seems to have much more of a holistic systems way of thinking in spirit, while being much weaker in the more theoretical, systematic approach to systems analysis, design, and implementation.

3 Linking Community Purpose to Functionality Usage with Collaboration Patterns

How to make sense of all this fragmentation and gaps between CI and IS, between research and practice, between rich, situated stories, and more systematically described, reusable lessons learnt? Patterns may help us chart our way forward.

3.1 Socio-Technical Pattern Languages

Patterns are a way of recognizing and describing approaches and structures that are encountered repeatedly in a discipline. They were first popularized by Christopher Alexander in an architectural context (Alexander, in (Dixon 2009). Patterns are descriptions of what exists and prescriptions for what should be done and supply direction for interpreting and acting on processes (Schuler 2008, 15, 55). They define relatively stable solutions to recurring problems at the right level of abstraction, making them concrete enough to be useful, while also sufficiently abstract to be reusable (de Moor 2006).

Patterns are organized in pattern languages, which are networks of patterns that call upon one another. Patterns help us remember insights and knowledge about design and can be used in combination to create solutions (Alexander in (Schuler 2002). Thus, patterns form a language by the way they are related to and used in combination with each other. Therefore, much more than being a mere collection of patterns, a pattern language is an *ecology* of patterns. Such a living knowledge base promotes, rather than restrains creativity, collaborative, and critical thinking, integrates theory and action and bridges traditional boundaries (Schuler 2008, 55, 543).

Socio-technical pattern languages are of particular interest to CI. As socio-technical systems are increasingly all around us, as ever more non-professionals understand these systems and their patterns, and a rich evolutionary base is emerging from which new patterns can emerge. These patterns are very similar to Alexander's patterns in that both focus on two interrelated systems being at work (i.e. Alexander's events vs spaces and our social and technical systems) and that both are strong on user involvement, using patterns to surface understanding in non-professionals (Dixon 2009).

According to Dixon, socio-technical patterns work at a different level of detail from more technical design patterns that focus on interface, interaction, and implementation, such as prominent in human-computer interaction and software engineering pattern languages (Borchers 2000). Instead, socio-technical patterns are especially useful at the beginning of social software projects to describe in a broad way – at the application domain level - the complex nature of the interactions between the social and technical systems that need to be built.

Too often, the focus is only on collecting and documenting patterns, instead of *using* them (Dixon 2009). As we stated, the main role of socio-technical pattern languages is to elicit requirements from stakeholders in the application domain. However, these specifications are not end products, but starting points for informing the systems design and implementation stages, including human-computer interaction and software engineering issues. Dixon outlines a practical approach that allows to go beyond just the technology and user interface. He describes how he has used workshops to identify and validate the right patterns, understand the relationships between the patterns and how these might reinforce one another, and finally to inform the design team of a computer-mediated communications system about the configuration of individual features and page design (Dixon 2009).

Socio-technical pattern languages are getting traction. Some stem from a software engineering or human computer interaction tradition, e.g. the interaction design patterns of Borchers (2000) and the socio-technical pattern language proposal of Thomas, Danis, and Greene (2002). Other languages focus more on putting specific types of tools like wikis in a (task-oriented) usage context, like the wikipatterns of Mader (2007). Other languages again focus more on defining the patterns for supporting particular social processes, such as group processes[2]. One that is closest to the community and social change tradition of CI, and most comprehensive in terms of number of patterns and the level of care with which they have been described, is the Pattern Language for Living Communication (Schuler 2002; Schuler 2008).

To give a flavour of the great variation in socio-technical pattern languages, we briefly characterize two that cover the full spectrum from community purpose to functionality usage-orientation: the Pattern Language for Living Communication and Wikipatterns.

3.2 Community Purpose Patterns: Pattern Language for Living Communication

The Pattern Language for Living Communication is a collection of socio-technical patterns that integrate theory and practice by showing how information and communication (face-to-face, broadcast, or Internet-based) can be used to address urgent social and environmental problems collaboratively. Each pattern contains a summary of the problem addressed, the context, a discussion, a solution, and a set of linked patterns. The pattern language is being developed in a long-term project by a network of authors. For submission, review, editing, presentation, and dissemination purposes, a central website is used[3]. Recently, a collection of 136 of the most mature patterns was published in a book (Schuler 2008). They include such diverse topics as Civic Intelligence, Social Responsibility, Opportunity Spaces, Citizen Science, Community Animators, The Power of Story and Tactical Media. Patterns are organized in several broad categories: Theory, Organizing Principles, Enabling Systems, Policy, Collaboration, Community and Organization Building, Self-Representation, Projects, and Tactics. Part of the network formed by these patterns is visualized in Fig. 2

[2] http://grouppatternlanguage.org

[3] http://www.publicsphereproject.org/patterns/

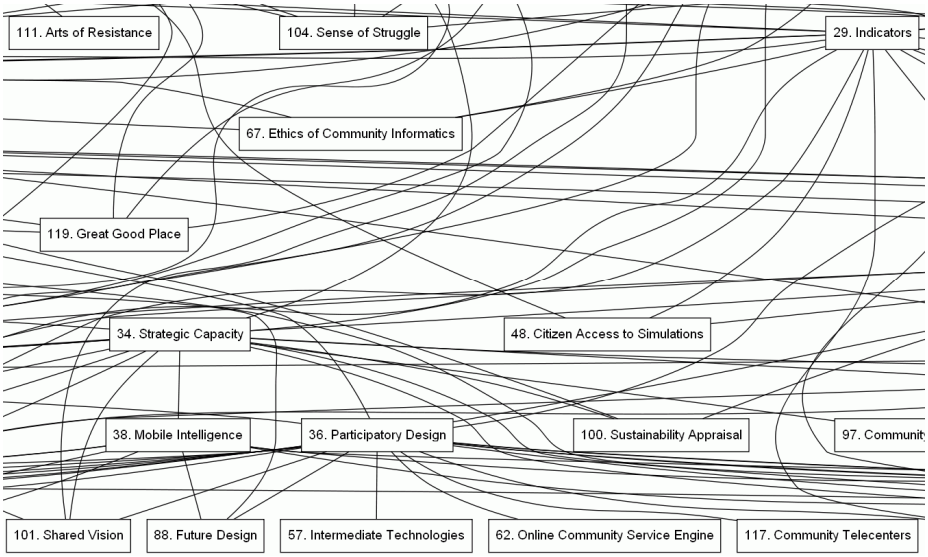

Fig. 2. The Liberating Voices pattern language (Schuler, 2008, our visualization)

The ultimate goal of the project is the construction of a large, structured collection of knowledge that represents the wisdom of a widely distributed, very loosely knit community of activists, researchers, policymakers, and technologists from around the world. According to Schuler, the patterns together could be considered an action-oriented (meta)theory of communicative space. This theory is about how information and communication can be used for positive social change. Each Liberating Voices-pattern is a mini-theory, with claims that using the ideas in the pattern can bring about positive social change in one way or another. The patterns show implicitly how the world works in the information and communication sphere, and form a collection of opportunities for social change. (Schuler 2008, 14, 536, 540). In sum, the Liberating Voices pattern language is a good candidate for capturing specific community purposeful contexts, values, cases and information/communication process requirements. The ecology of patterns, the online system and the community of authors and users around this pattern language ensure that the why, what, and who of the required community systems is (informally) specified and grounded in the larger CI community of researchers and practitioners.

The next goals of the project are to strengthen the pattern language community, improve the comprehensiveness and overall effectiveness of the pattern language, and improve the online environment to support these objectives. Services in mind include repository services, an exploratory environment, a collaborative learning space, annotation capacity, and collaborative testbed development, among other things (Schuler 2008, 532-4). These goals, necessary as they are, focus on the further development of the pattern language itself. However, Schuler makes an important observation when he says that the main users of patterns are not necessarily the ones who have looked at the patterns directly, but those who may be using a service or

information resources developed by somebody who has (Schuler 2008, 28). In other words, these patterns could have much more impact when informing the *design of socio-technical systems.* To do so, we need to look beyond this particular pattern language.

3.3 Functionality Usage Patterns: Wikipatterns

The Liberating Voices pattern language is strong in defining contexualized and value-based requirements, connecting CI theory and practice. Still, it provides many degrees of freedom in defining the *how* in terms of the selection, configuration, and use of the specific technological functionalities needed to satisfy these requirements. A plethora of technologies has evolved, from ancient small-group face-to-face meetings to an ever more advanced array of Web 2.0 tools. Which of these tools can best satisfy particular community information and communication needs? What are best (or even *good* practices[4]) of their use in particular contexts? Another class of pattern languages is emerging which studies the use practices of specific technologies, including the user roles, permitted actions, information objects used and produced, and conditions for the creation and modification of these objects. One of the most developed pattern languages in this respect is the Wikipatterns language, which consists of a living knowledge base of lessons learnt about wikis in collaborative contexts of use (Mader 2007).

Each wikipattern consists of its name, a summary of its intent, usage, an example, related patterns, and further reading. A community of authors uses – of course – a wiki to develop the patterns[5]. Patterns are classified along two dimensions: whether they are "people" or "adoption" and whether they are "(desired) patterns" or "antipatterns", describing behaviour that should be prevented. Examples of people patterns are Ambassador, WikiGardener, and WikiZenMaster. People anti-patterns include Bully, OverOrganizer, and WikiTroll. Some adoption patterns are Agile Lifecycle, Community Portal, and Critical Mass. Some examples of adoption antipatterns are Build It And They Will Come, Design By Committee, and Too Much Structure.

3.4 Designing the Socio-Technical Community System: the Need for Collaboration Patterns

As stated before, pattern languages are not developed for their own sake, but to be used and have an impact in practice. They should help describe coherent, actionable knowledge in complex domains (Schuler 2008, 522). We have introduced the Liberating Voices pattern language as an example of a source of community-based, action-oriented information and communication *requirements* (why, what, and who is to be involved in the information and communication processes?) and the Wikipatterns language as a representative of effective *potential* use of particular information and communication technologies (how are these processes to be

[4] Acknowledgments to Rolf Kleef for making this observation.
[5] http://wikipatterns.com

effectively supported?). It is obvious that in the design of effective, action-oriented socio-technical systems that enable the *actual* context, value, process, case, and system requirements of grounded, *particular* communities we need both. How are they to be combined, however?

To answer this question, a third class of patterns is needed: *collaboration patterns* (Fig. 3). First, these patterns should be able to represent the socio-technical systems in which the required information and communication processes of a particular community are *supported* by a specific set of functionalities. Second, these patterns should capture the triggers for *activating* the communities using these systems so that the overall community purpose is fulfilled (de Moor 2008).

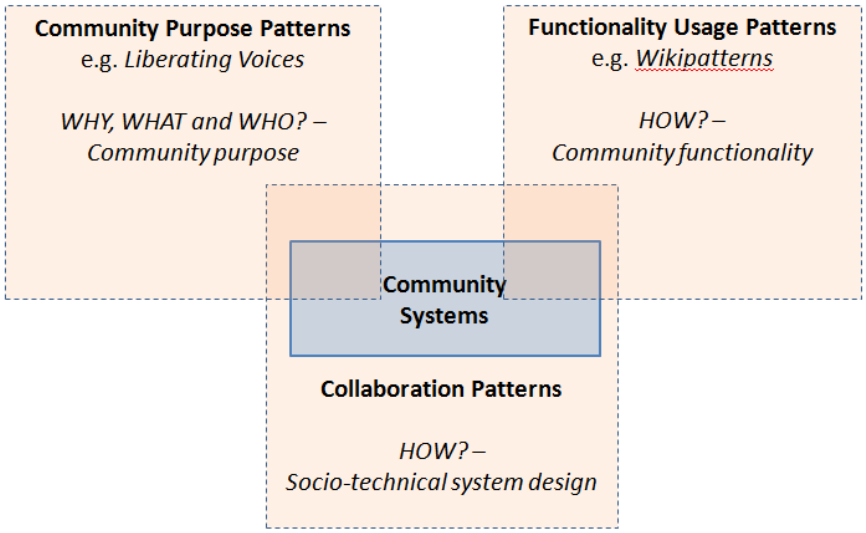

Fig. 3. Designing the socio-technical systems of communities with collaboration patterns

Why call them *collaboration* patterns? Collaboration patterns capture socio-technical lessons learnt in optimizing the effectiveness and efficiency of collaboration processes (de Moor 2006). They tie community purpose to technical functionalities. Community systems not only enable information processes and communication processes to exchange the information produced and to create a sense of community (Koh and Kim 2004). Most communities are also collaborative, in the sense that goals must be realized and tasks must be coordinated. Goals can range from writing an article via setting up a local innovation community to changing the world in global, concerted e-campaign. Coordination is the act of managing interdependencies between activities performed to achieve a goal. There are many kinds of interdependencies to be managed, from shared resources to task and subtask dependencies (Malone and Crowston 1994). Collaboration patterns now describe how the socio-technical system should enable all collaboration processes (information,

communication, and coordination processes) that are necessary and sufficient for accomplishing the the overall community purpose.

The community system should provide a full range of collaborative and communicative functionalities (Preece 2000; Wershler-Henry and Surman 2001). Collaboration patterns make these functionalities *actionable* by describing how individual community members playing particular collaborative roles use particular functionalities. Making such activation triggers explicit, is a necessary condition for community members to grow from being mere readers, to first becoming contributors, then collaborators, and sometimes even leaders (Preece and Shneiderman 2009).

To serve these purposes, collaboration patterns include goal, communication, information, task, and meta-patterns. In (de Moor 2006; de Moor 2007; de Moor 2008), we gave a detailed description of our typology of collaboration patterns, the ontology to conceptualize the usage context, tool system, and socio-technical mappings between these subsystems, and the way in which to obtain, represent, and use these patterns to model the socio-technical system and its activation. We do not have the space to reintroduce all these elements of our methodology here. For more details, we refer the reader to these papers. However, we next outline our approach with a scenario based on a global collaborative case.

4 Collaboration Patterns in Practice: The ESSENCE Scenario

We first describe the purpose of the ESSENCE project, then frame it in socio-technical system terms, and continue by showing how a pattern analysis can help in the design of such a system.

4.1 The Purpose of ESSENCE

The ESSENCE (E-Science/Sensemaking/Climate Change) project is the first major project that emerged out of the GlobalSensemaking Network[6]. In this project, several developers of leading-edge Web-based argumentation mapping and "sensemaking" tools are collaborating to make their tools more useful and interoperable (see (Buckingham Shum 2007) for an introduction of these tools). The question is how to achieve "actionable sensemaking"? To answer this question, the project has a set of ambitious, but very important and interlinked goals:

- **pilot software tools designed to help facilitate structured analysis and dialogue**: map, summarise and navigate science and policy arguments underpinning the issues that people care about and collaborate with the development teams who are shaping these tools to meet the complex, systemic challenges of the 21st Century.
- **develop a comprehensive, distilled, visual map of the issues, evidence, arguments and options** facing the UN Climate Change Conference in Copenhagen, and being tackled by many other networks, which will be available for all to explore and enrich across the web.

[6] http://globalsensemaking.wik.is/ESSENCE

- **build a definitive, public collective intelligence resource** on the climate change debate.
- **support dialogue** that builds common ground, resolves conflict, and re-establishes trust.

4.2 A Socio-Technical System Perspective

In the ESSENCE approach, the sensemaking tools are not examined in splendid technical isolation but as embedded in living, growing socio-technical systems.

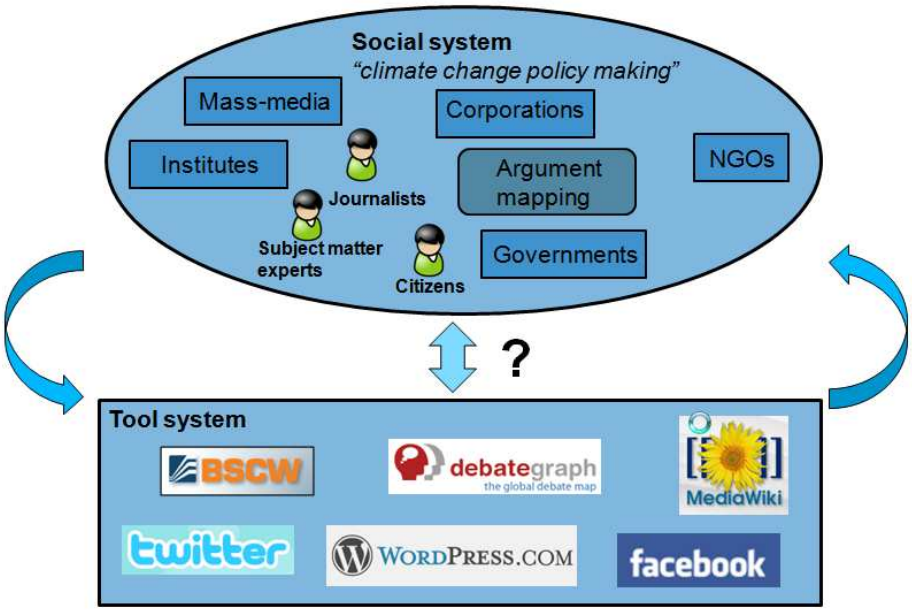

Fig. 4. A socio-technical systems view on ESSENCE

An example of what such a socio-technical system could look like, a (hypothetical) scenario which could play out in the case of ESSENCE is given in Fig. 4. In broad strokes, the system consists of a social system, where the main purpose is to improve climate change policy making. The main workflow currently concerns the mapping of the key arguments in the global climate change debate. Actors representing a great number of domain stakeholders are involved in the community, including corporations, governments, NGOs, research institutes, and the mass media. Also, many individuals have joined this community, including subject matter experts, journalists, and concerned citizens. The technical system supporting the community

consists of a number of tools. At the core is a sensemaking tool, Debategraph[7], which is used to represent and visualize key climate change issues, positions, and arguments. This tool is embedded in a cloud of information and communication tools: Facebook for informal community discussion, Wordpress for blogging about new developments in the community, BSCW for document management, Twitter for announcing updates and events, and a MediaWiki for co-authoring community documents. Both the social and technical system are heavily in flux and co-evolve all the time.

4.3 Defining Requirements with Community Purpose Patterns

To find practical ways to operationalize the community purpose we turn to the Liberating Voices Pattern Language. One relevant pattern seems to be the *Citizen Journalism* pattern[8]. This pattern states that the traditional mass media fail to inform citizens about acting effectively on societal problems. It suggests that the Web provides numerous opportunities for citizens to become more actively involved in the news stream, from the local all the way up to the global level, and from fact-based to opinion-based journalism. All of these are important inputs for climate change policy making. By participating directly in the production and dissemination of journalism, citizens can help (re)set the news agenda. According to the pattern, challenges to be aware of include, first, that a broader community dialogue should emerge, not to have it dispersed into many small communities of group monologue, and, second, that citizen journalism should ensure to be powerful enough to have an impact. The first challenge is addressed as the argument mapping support through Debategraph ensures focus[9], the second by having many stakeholders, not just citizens involved in this community.

Inspired by this pattern, the community decides to develop a *Public Investigator*-service, which should give the general public a larger and more informed say in the climate change debate. One design for such a service could be showing a simple web page that lists controversial issues with their attached positions, arguments and associated stakeholders and provides clickable links to supporting contact information and documentation. This service could be especially useful for individuals, say concerned citizens or investigative journalists (de Moor, Park, and Croitoru 2009). This way forward looks promising. However, the community decides to go into a different direction, at least initially, as it wants to focus more on the collaborative instead of the individual aspects of this service.

Up to now, the argument mapping workflow of the climate change community consisted of the mapping of issues, positions, and arguments pro and con. However, within this community of passionate and knowledgable members, there is an increasing desire to *use* these maps to co-author well researched group reports with

[7] http://debategraph.org/

[8] http://www.publicsphereproject.org/patterns/pattern.pl/public?pattern_id=804

[9] Strictly speaking, the selection of Debategraph as the particular tool to be used should take place in the pattern analysis of the functionality usage. However, in reality, many technological constraints will have been defined up front. The functionality usage analysis is then still useful. It can help to further constrain this space by defining the most relevant potential technologies to be used in a particular collaborative setting.

policy recommendations. Key community *values* to be taken into account in the design of such a GRASS (Group Report Authoring Support System) are *neutrality* and *transparency* (Heng and de Moor 2003). Neutrality does not mean that such a report describes a single, meaningless, "forced consensus" position. Rather, it should be a true dialogic text reflecting multiple authorial voices (Harrison and Stephen 1992). Transparency means that the full process in which the current version of a text came into being can be retraced.

We next need to look how to best use particular functionalities to ensure that these complex requirements are met.

4.4 Selecting Relevant Technologies with Functionality Usage Patterns

Wikis are very suitable for collaborative editing purposes of complex, multi-authored, versioned documents. The general idea would be to first embed in the wiki parts from the Debategraph argument map on climate change relevant to the topic of the report to be written. Every web part should gets its own wiki page. Each of these wiki pages should contain a section containing the recommended consensus position, plus any number of sections describing the alternative positions that only a minority of the report authors subscribe to. Now the Debategraph argumentation map and the wiki page can co-evolve: as new Debategraph arguments become available, policy recommendations on the wiki can be adapted. Vice versa, as recommendations (and thus implications) get discussed on the wiki, new ideas will emerge, and stakeholders may readjust the issues, positions, and arguments outlined on the argumentation map. Thus, by combining the relative strengths of both tools (Debategraph for structured argumentation, wiki for collaborative document production), activation can be increased.

As they preserve the "creation trail" of wiki pages, the transparency value mentioned above can be guaranteed. However, what about neutrality? How to ensure that the use of the wiki takes place in the neutral way desired? As neutrality is a very fragile notion, which can easily be destroyed, trust and mutual respect among adversaries is of the essence. To get some ideas about best practices here, the community studies the Wikipattern language[10]. Three wikipatterns that seem especially relevant in this initial stage are BarnRaising[11], Wiki Charter[12], and Maintainer[13].

BarnRaising is a planned event in which a community meets at a designated time to build content on the wiki together. In this activity, people come expecting to learn how to use the wiki, through these initial interactions help strengthen community bonds and create a support network that keeps people using the wiki. Such bonding is also helpful for stakeholders with conflicting interests to establish a minimum level of trust and goodwill towards each other, especially necessary in our case.

The *Wiki Charter* is a document that sets guidelines for community collaboration and respectful, productive activity on the wiki. It is important to have such a

[10] http://wikipatterns.com
[11] http://wikipatterns.com/display/wikipatterns/BarnRaising
[12] http://wikipatterns.com/display/wikipatterns/Wiki+Charter
[13] http://wikipatterns.com/display/wikipatterns/Maintainer

document right from the start, especially in mostly online communities like on climate change.

A *Maintainer* is a person (self-)assigned to a page, space, or section of a wiki who accountably takes responsbility for the quality of some content. The role may include that of *secretary* (collecting information from comments and meetings into the wiki), *refactorer* (collapsing redundancy and inserting organization into a wiki), *solicitor* (encouraging input from community members), and *architect* (categorizing pages, creating 'project' and 'overview' pages, and assigning meaning to labels).

Based on the barnraising concept. the community decides to hold a kickoff meeting with representatives of all stakeholders in the climate change community. After that meeting, a brief charter will be written by one of the proposed report editors about the rights and responsibilities of the report editors, the section editors, and the authors. The maintainer role, in all its four subroles, will be used in the definition of the collaboration patterns that the report and section editors are involved in.

4.5 Designing the Socio-Technical System with Collaboration Patterns

We now know our requirements through the community purpose pattern analysis (group report authoring process resulting in sections describing consensus and alternative positions on an issue) and the most relevant technologies to be used (Debategraph and wikis as primary group report authoring tools. The next step is to complete the design the socio-technical system. We define this system as consisting of the integrated whole of requirements (goals, roles, objects, and workflows) and enabling functionalities specified, all specified at the right level of detail, so that the community purpose can be effectively and efficiently achieved. In doing so, particular attention needs to be paid to activation issues.

We distinguish five types of collaboration patterns, slightly adapted from (De Moor 2006):

- **Goal patterns** are conceptual representations of community and individual objectives.
  Example: a community objective could be to produce a policy report addressing one key question, and containing one consensus position and at least two alternative positions. An individual objective could be to describe one such an alternative position.
- **Information patterns** are conceptualizations of content knowledge essential for the collaborative process plus the actors responsible for their creation and maintenance.
  Example: each policy report consists of an embedded argument map maintained by external stakeholders interested in climate change, a section describing a consensus position maintained by a report editor, and one or more sections describing alternative positions to be maintained by section editors.
- **Communication patterns** are sets of communicative workflow and norm definitions describing acceptable and desired communicative interactions within a community.

Example: once the report editor updates the consensus position, she must notify all the other authors of the report. Each of them may decide to challenge the updated consensus position. The report editor must then redefine the consensus position until no more objections are made.

- **Task patterns** define which information patterns in a communication process are to be created and exchanged when, by whom, and through what particular steps, thus completing the workflow definitions required by a communication process.

  Example: position updates (whether of consensus or alternative positions) must be posted on a publicly available common information resource. Any challenge of such an update must be made through a threaded comment facility to which all report authors have access, so that it is clearly visible who replies to what argument. A daily digest is to be sent to all authors with the status updates of that day, plus a ranking of the Top-5 positions that have been most heavily discussed.
- **Meta-patterns** are conceptual patterns necessary to interpret, validate, link, and assess the quality of other collaboration patterns.

  Example: a "neutrality-pattern" could define that all authors must have the right to initiate any communication-pattern concerning the production of a group report.

Core social system requirements to be modeled are goals, actor roles, information objects, the information and communication processes in which these objects are produced and exchanged, the tasks in which the production of these objects is coordinated, and the quality aspects that link and constrain the combinations of these processes. Together, these are classified *as required collaboration patterns*.

An example is the required information pattern describing the structure of a policy report (Fig.5)[14], as presented in the information pattern example above:

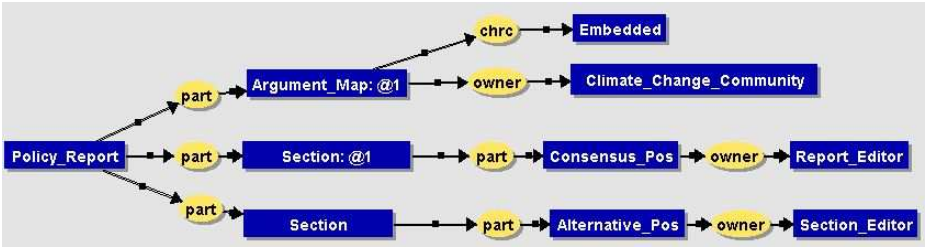

Fig. 5. A required information pattern

Enabled collaboration patterns define the mappings between the social and the technical system as they specify how particular required collaboration patterns are

[14] This representation is in conceptual graphs notation. Conceptual graph theory is a powerful formalism for knowledge representation and reasoning that is grounded in linguistic principles on the one hand, and formal semantic network representations on the other hand. As such, it is very well suited for playing a bridging role between social and technical pattern languages. See (De Moor 2006) for more explanation and examples.

enabled by specific functionalities of the technical system. To define these functionalities at the right level of detail[15], we distinguish four different functionality levels: the *system* level (the total set of tools used, e.g. the group report authoring system), the tool level (e.g. Debategraph or Mediawiki), the *module* level (a logical grouping of functions within a tool, e.g. calendar, page editing functions, mail functions), and the *function* level (e.g. sending, copying, or forwarding a mail).

An example is the following enabled communication pattern. It describes how the communication process around the kickoff meeting is supported by the technical system (Fig. 6). It shows how the output of this process is a policy report charter, to be created by one of the proposed report editors. The kickoff is started by this same (indicated by the dotted line) report editor, using whatever tool she prefers[16]. The kickoff is performed by selected representatives of the "climate change community". It is agreed that everybody is to use the videoconferencing module of Skype for this purpose. Finally, once the draft policy report charter has been prepared by the report editor, the representatives are to discuss and approve it using their mailing list.

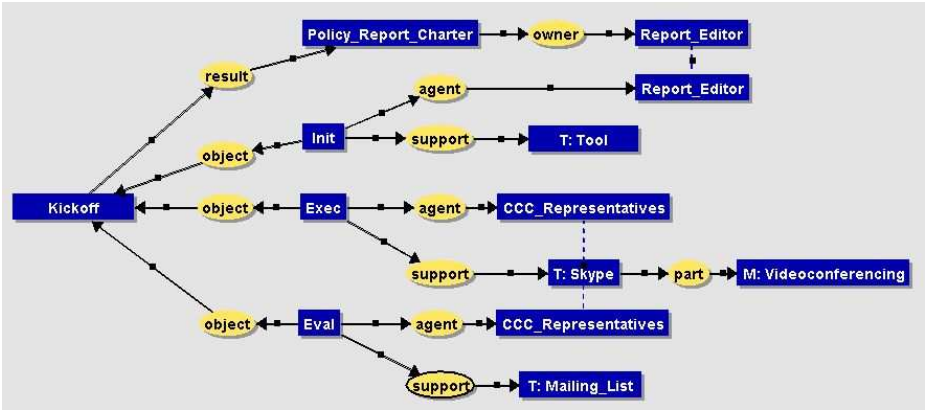

Fig. 6. An enabled communication pattern

We gave only a brief illustration of what such required and enabled collaboration patterns look like and how they are to be used. The papers mentioned earlier describe the methodology in more detail. What is important to realize is that these patterns are linking pins between the informal requirements world of community members (as expressed by community purpose and functionality usage patterns) and the more formal design/implementation worlds of the human computer interaction and software engineering patterns worlds. It is our contention that by combining these patterns in intelligent ways, community systems design can be made more effective and take

[15] Sometimes it is enough just to know that a particular tool such as a wiki is to be used, in other cases we need to know exactly which module or function of such a tool should be employed.

[16] By alllowing for tools and other entities to be defined at different levels of specificity, collaboration patterns are powerful representations for describing necessary and sufficient constraints on socio-technical systems.

place more efficiently, as lessons learnt are increasingly reused and recombined instead of forgotten.

5 Discussion and Conclusions

In this paper, we made the case for collaboration patterns as being essential building blocks for CI. These patterns are key to informing the analysis, design, implementation, and evaluation of socio-technical community systems, and thus at the heart of CI research and practice. We had to cover a lot of ground, and this paper has already become much longer than intended. Of necessity, we had to leave out many details. Also, many implications of the approach proposed are only barely visible, let. Still, to get the discussion started, we list at least some of the many issues to be addressed in future work.

Socio-technical pattern languages are emerging that help capture, reflect upon and disseminate good and best practices in the purposeful, informal requirements world of communities and the formal design and implementation world of the technical systems that are to support them in their work. However, there is still a wide "socio-technical divide" between both worlds. One of the main contributions of the paper is to link community purpose with functionality usage pattern languages through collaboration patterns. Collaboration patterns link and formalize these other patterns with a focus on effective tool systems for and activation of the communities of use. In our earlier work, mentioned above, we outlined the ontology for the conceptualization of and an approach for eliciting and applying collaboration patterns. Much more work is needed, however, to develop this into a mature methodology. Also, work still has to start to research how this emerging collaboration pattern language could in turn be inform existing human computer interaction and software engineering pattern languages (cf. Dixon 2009). The latter types of languages are needed to design and develop robust and usable implementations of the collaborative systems.

We focused on Liberating Voices as a representative of a community purpose pattern language, and on Wikipatterns as an example of a functionality usage pattern languages. However, there are many other candidates which could be mined to create a base of "reference" patterns. Liberating Voices represents the "soci(et)ial change" tradition in CI. A (closely related) stream would be work done in Development Informatics, which, although also working on social change, has more eye for practical ICT resource problems and thus may be a more comprehensive source for functionality usage patterns. Another class of CI-related patterns could draw from the communities of practice, knowledge management etc. tradition developed by the more organizational and technological inclined CI researchers and practitioners, like (Wenger 2002; Preece 2000; Bieber et al. 2007).

Socio-technical pattern languages are not static knowledge representations, but living organisms, which each have their own communities of interest/practice around them for evolving them. The collaboration pattern language community is still in its infancy. Given that these pattern languages and their communities are distinct yet interconnected, it might be worthwhile starting a "meta-community" in which these communities try to align their work, prevent redundancy, exchange best practices, develop interfaces between the languages, and so on. One benefit would be more

stimulus and motivation to commence and continue work on developing individual patterns, as potential applications are often discovered outside a particular pattern language community. Thus, what we propose is an "ecology of pattern *languages*". To help inter-language collaboration, a formal "lingua franca" might be developed to express and link community purpose, functionality usage, collaboration, human computer interaction, and software engineering patterns. Conceptual graph theory seems a very promising direction to study in this respect, but alternative knowledge representation formalisms can be used as well, of course. By more systematically clarifying and connecting patterns throughout this full socio-technical systems development process, more robust and effective socio-technical systems of higher quality can be developed faster.

Technology is a very loaded and sensitive term in CI. It affords, but can also constrain, suffocatingly so. It can help build bridges, but also create disempowering (digital) divides. Too often, ICTs have been interpreted in the very narrow, high-technological, or infrastructural sense. The socio-technical pattern language approach outlined in this paper should help cure this myopia. Technology should be regarded as any means -in the broadest possible sense- by which the community end can be served. A face-to-face meeting in a remote village is one of the most powerful technologies possible, but has often been overlooked as such in traditional information systems development theory and practice. A collaboration pattern-way of thinking and working on the other hand, is not a rigid, reductionist engineering approach. It is a communal sensemaking exercise, allowing critical reflection on the affordances and constraints provided by the plethora of technologies now at humanity's disposal. Collaboration is about human beings working together on joint purpose. By studying and (re)using patterns of effective collaborative use of the power of technology, we may get one step closer to building better socio-technical systems. These are after all, although not sufficient, the necessary catalysts for bringing about a somewhat better world.

References

Bieber, M., B.S. McFall, R.E. Rice, and M. Gurstein. 2007. Towards System Design for Supporting Enabling Communities. *Journal of Community Informatics* 3, no. 1.

Borchers, J.O. 2000. Interaction Design Patterns: Twelve Theses. In *Pattern Languages for Interaction Design: Building Momentum, CHI 2000, The Hague, the Netherlands, Apri 2-3, 2000*.

Buckingham Shum, S. 2007. Hypermedia Discourse: Contesting Networks of Ideas and Arguments. In , 4604:29-44. Springer.

Day, P. 2005. Sustainable Community Technology: The Symbiosis Between Community Technology and Community Research. *Journal of Community Informatics* 1, no. 2.

Day, P., and D. Schuler. 2004. Integrating Practice, Policy and Research. In *Community Practice in the Network Society: Local Action, Global Interactions*, ed. P. Day and D. Schuler. NY: Routledge.

Dixon, D. 2009. Pattern Languages for CMC Design. In *Handbook of Research on Socio-Technical Design and Social Networking Systems*, ed. B. Whitworth and A. de Moor, 402-415. Hershey PA: Information Science Reference.

Gurstein, M. 2008. Community Informatics: What's in a Name? *Journal of Community Informatics* 4, no. 3.

Gurstein, M., and T.A. Horan. 2005. Why Community Information Systems are Important to the Future of Management Information Systems and the Field of Information Science (IS). In *The Gordon B. Davis Symposium: The Future of the Information Systems Academic Discipline - Opportunities and Directions*. University of Minnesota.

Harrison, T.M., and T. Stephen. 1992. On-Line Disciplines: Computer-Mediated Scholarship in the Humanities and Social Sciences. *Computers and the Humanities* 26: 181-193.

Heng, M., and A. de Moor. 2003. From Habermas's Communicative Theory to Practice on the Internet. *Information Systems Journal* 13, no. 4: 331-352.

Koh, J., and Y.-G. Kim. 2004. Sense of Virtual Community: A Conceptual Framework and Validation. *International Journal of Electronic Commerce* 8, no. 2: 75-93.

Mader, S. 2007. *Wikipatterns*. Wiley.

Malone, T.W., and K. Crowston. 1994. The Interdisciplinary Study of Coordination. *ACM Computing Surveys* 26, no. 1: 87-119.

de Moor, A. 2006. Community Memory Activation with Collaboration Patterns. In *Proc. of the 3rd Prato Community Informatics Research Network Conference (CIRN 2006), Prato, Italy, October 2006*.

de Moor, A. 2007. The Pragmatic Evaluation of Tool System Interoperability. In *Proc. of the 2nd ICCS Conceptual Structures Tool Interoperability Workshop (CS-TIW 2007), Sheffield, UK, July 22, 2007*, 1-19.

de Moor, A. 2008. Activating Online Collaborative Communities. In *Proc. of the 5th International Conference on Action in Language, Organisations, and Information Systems (ALOIS 2008), Venice, Italy, May 5-6, 2008. University of Trento*, 97-108.

de Moor, A. 2009. Moving Community Informatics Research Forward. *Journal of Community Informatics* 5, no. 1.

de Moor, A., and M. Aakhus. 2006. Argumentation Support: From Technologies to Tools. *Communications of the ACM* 49, no. 3: 93-98.

de Moor, A., J. Park, and M. Croitoru. 2009. Argumentation Map Generation with Conceptual Graphs: the Case for ESSENCE. In *Fourth Conceptual Structures Tool Interoperability Workshop (CS-TIW 2009), Moscow, July 2009*.

Preece, J. 2000. *Online Communities : Designing Usability, Supporting Sociability*. Chichester ; New York: John Wiley.

Preece, J., and B. Shneiderman. 2009. The Reader-to-Leader Framework: Motivating Technology-Mediated Social Participation. *AIS Transactions on Human-Computer Interaction* 1, no. 1: 13-32.

Schuler, D. 2002. A Pattern Language for Living Communication: A Global Participatory Project. In *Participatory Design Conference (PDC'02), Malmo, Sweden, June*.

Schuler, D. 2008. *Liberating Voices: A Pattern Language for Communication Revolution*. Cambridge, Mass: MIT Press.

Stillman, L., and H. Linger. 2009. Community Informatics and Information Systems: Can They Be Better Connected? *The Information Society* 25, no. 4: 255-264.

Stoecker, R. 2008. Creative Tensions in the New Community Based Research. In *Researching with Communities: Grounded Perspectives on Engaging Communities in Research*, ed. A. Williamson and R. deSouza, 7-21. Muddycreekpress.

Thomas, J. , C. Danis, and S. Greene. 2002. Socio-Technical Pattern Language Proposal. In *CSCW 2002, New Orleans, November 16-20, 2002*.

Wenger, Etienne. 2002. *Cultivating Communities of Practice: A Guide to Managing Knowledge*. Boston, Mass: Harvard Business School Press.

Wershler-Henry, D., and M. Surman. 2001. *Commonspace: Beyond Virtual Community*. FT.Com Books. Toronto: Pearson.

Whitworth, B. 2006. Socio-Technical Systems. In *Encyclopedia of Human Computer Interaction*, ed. Claude Ghaoui, 559-566. Hershey PA: Idea Group Reference.

Whitworth, B. and A. de Moor, eds. 2009. *Handbook of Research on Socio-Technical Design and Social Networking Systems*. Hershey PA: Information Science Reference.

Williamson, A. 2008. Introduction. In Williamson, A. and DeSouza, R. (eds.), *Researching with Communities: Grounded Perspectives on Engaging Communities in Research*, 1-6. Muddycreekpress.

Williamson, A., and R. DeSouza, eds. 2007. *Researching with Communities: Grounded Perspectives on Engaging Communities in Research*. Auckland: Muddycreekpress.

Yin, R. K. 1994. *Case Study Research: Design and Methods*. 2nd ed. Applied social research methods series v. 5. Thousand Oaks: Sage Publications.