Towards a Design Theory for Community Information Systems[1]

Aldo de Moor

STARLab, Vrije Universiteit Brussel
Pleinlaan 2, Building G-10, 1050 Brussels, Belgium
ademoor@vub.ac.be

Abstract

Virtual communities are complex and evolving socio-technical systems. The design of community information systems requires much theoretical research to solve design problems. A design theory is a prescriptive theory which helps optimize design methods. Community IS design theory is still very young and fragmented, however. In the design theory development process, a mix of theory components is used to solve novel problems or existing problems more effectively and efficiently. We present a meta-model of IS development which focuses on the role of theory in IS design. We show how simulation via system dynamics could play an important role in a more systematic development of design theory for community information systems.

1 Introduction

Virtual communities are complex and evolving socio-technical systems. Communities are not just aggregates of people, temporarily interacting. They have been defined as groups of people who share social interactions, social ties, and a common space (Kozinets, 1999). A virtual community differs from other communities in that its common space is cyberspace. Virtual communities therefore describe the union between individuals or organizations who share common values and interests using electronic media to communicate within a shared semantic space on a regular basis (Schubert and Ginsburg, 2000).

Much research on online communities has focused on community characteristics like success factors, drivers, roles, and social norms (e.g. Andrew, 2004). Other research examines the role of specific technologies, for example, the impact of specific technologies on community performance (e.g. Schubert and Koch, 2003). Yet another major stream of research studies the process of technology adoption by communities, and what happens in case of a lack of technologies, such as problems with respect to effective use and the Digital Divide (e.g. Gurstein, 2003).

What much of this research has in common is that it studies partial aspects of the socio-technical systems that virtual communities are. Another important line of research, however, concerns the *process* in which the *information systems* of virtual communities can be developed. In this development process, a mix of technologies is tailored to the unique needs of a particular online community. Some key approaches addressing this problem are Wenger et al.'s (2002) approach to cultivating communities of practice, Gongla & Rizzuto's approaches to evolving communities of practice (2001), and Preece's (2000) method for community-centred development. Still, systematic methods for community information systems development solidly grounded in theory are rare and only premature.

IS research is still a young discipline. Paradigms, concepts, models, methods and techniques are still in the early stages of development. Much IS development theory is inspired by a behavioral science paradigm, in which the goal is to develop and justify theories that explain or predict organizational and human *phenomena* surrounding the analysis, design, implementation, management and use of information systems. A complementary perspective, however, is provided by a problem-solving design science paradigm. It seeks to create *innovations* that define the ideas, practices, technical capabilities, and products through which the analysis, design, implementation, management, and use of information systems can be effectively and efficiently accomplished (Hevner et al., 2004). When trying to build theory on community IS development, this design science paradigm is an important starting point.

A design theory is prescriptive theory based on theoretical underpinnings which says how a design process can be carried out in a way which is both effective and feasible (Walls, 1992). Such a theory can help describe,

[1] Proc. of the 11th International Conference on Human-Computer Interaction (HCII 2005), Las Vegas, 22-27 July 2005. Lawrence Erlbaum Ass.: Mahwah, NJ.

analyze, and optimize design methods. To aid in theory development, Hevner et al.'s (2004) framework for conducting, evaluating, and presenting IS research is useful. The framework focuses on the relations between behavioral and design research processes. These processes are informed by both the ‘Environment’ (people, organizations, and technologies) and a ‘Knowledge Base’ of conceptual Foundations and Methodologies potentially useful in conducting the research. Using such an analytical framework can help researchers to develop new representations of IS problems, solutions, and solution processes.

Although such approaches are a useful starting point for analyzing community IS development, they are too generic. Community IS development has unique properties, such as the importance of effective communal use in terms of collaborative and collectively identified goals, the need to support sociability, and the strong interrelationship between theory and practice (Gurstein, 2003; Preece, 2000). Other design theories for specific classes of IS have been developed, such as a design theory for vigilant EIS (Walls et al., 1992) and for emergent knowledge process support systems (Markus et al., 2002). Our aim is to work towards a design theory specific for community IS. One element in particular we consider to be essential: the use of *simulation* as a way to mediate between theory and practice, as we will discuss later.

A starting point for a design theory of community IS is our meta-model of IS development. Based on Walls et al.’s and Hevner et al.’s approaches, we propose a meta-model of IS development, which focuses on the relationship between IS and IS design theory development. We then show how simulation could play an important role in theory formation when embedded in such a meta-model

In Sect.2, we examine the concept of IS design theories in greater detail. Sect. 3 introduces system dynamics theory as a powerful way of simulating complex and dynamic systems. In Sect. 4, we show how system dynamics can be useful in supporting theory development on community IS design. We end the paper with conclusions.

2 IS Design Theories

Information sytems are more than just arbitrary sets of hardware, software, and data. They are *systems* composed out of these elements whose mission it is to improve the performance of people in organizations through the use of information technology (McNurlin & Sprague, Jr., 1989). Typically, systems are developed in a number of stages. There are many different subdivisions of the systems development process. One widely accepted interpretation of two of its key stages is the folllowing: *systems analysis* is the process of collecting, organizing, and analyzing facts about a particular IS and the environment in which it operates. *Systems design* then is the conception, generation and formation of a new system, using the analysis results (Hirschheim et al., 1995, p.11). Frequently, there is an overlap between the two stages, but overall, the analysis stage focuses on what the information system must do, and the design stage on how the system will do it (Yeates et al., 1994).

The purpose of a design theory is to support the achievement of goals, contrary to a natural science theory. Walls et al (1992) give a good introduction to design theories for information systems. They show how explanatory, predictive, or normative theories can be put to practical use. In their view, design needs to be seen both as a process and as a product. From a product-perspective, a design theory consists of several components: *meta-requirements*, describing the class of goals to which the theory applies; *meta-design*, a class of artifacts hypothesized to to meet the meta-requirements; *kernel theories* governing design requirements; and *testable design product hypotheses*, to check whether the meta-design satisfies the meta-requirements.

From a process-perspective, the components are a *design method* describing procedures for artifact construction; *kernel theories* governing the design process; and *testable design process hypotheses*, to check whether or not the design method results in an artifact consistent with the meta-design.

Hevner et al. (2004) present a related view on IS design theories. In their information systems research framework, design science (applied to address novel organizational problems) creates and evaluates IT *artifacts* intended to solve the identified problems. IS research is influenced by the Environment of use (people, organizations, and technology), as well as by a Knowledge Base of theoretical components consisting of “Foundations” and “Methodologies”. To build IS relevant to the environment, applicable knowledge from the Knowledge Base must be applied in the building of artifacts that are part of the IS. These artifacts, put to use, must then be evaluated according to the utility criterion of how well they meet the business needs of the users. If the intervention has proven to be successful, the knowledge that a particular foundational/methodological component was useful in the design of a particular artifact for this particular context, can be added to the knowledge base. In their view, a clear distinction therefore must be made between “routine design” and “design science research” in which unsolved design problems are addressed in innovative ways, or solved problems in more effective and efficient ways (Hevner et al., 2004).

2.1 A Meta-Model of IS Development

Inspired by the previous perspectives, we present a meta-model of IS Development (Fig.1). The focus of this model is not IS development of a particular IS, but *theory development* of the design of IS. This model will be the basis for the specific approach to theory development on community IS design that we will propose in Sect. 4.

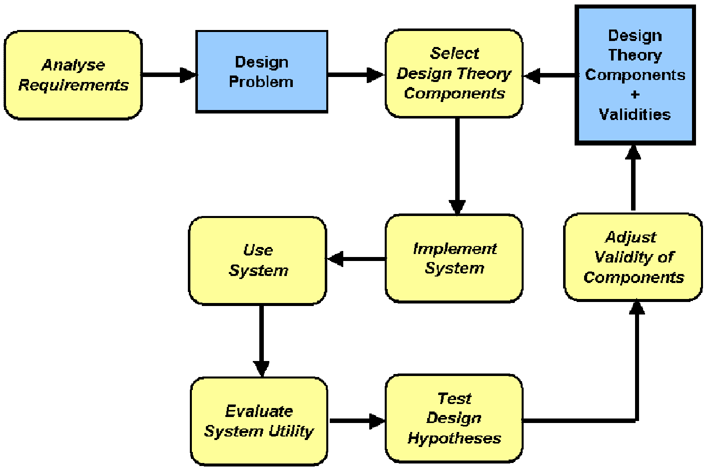

Figure 1 A Meta-Model of IS Development

At the heart of our IS design theory development are *theoretical components*[2], which comprise every conceptual or methodological element that can contribute to solving a design problem. Design theory development starts when requirements analysis establishes the need for addressing design problems in a novel or more effective/efficient way. Out of the knowledge base of design theory components, the systems developer selects those components that are most likely (i.e. have the highest validity) to help solve the problem. To capture the rationale for this decision, the designer formulates one or more design hypotheses in which is stated how the theory component could contribute to the resolution of the problem. After implementing and using the system, the information system utility is evaluated. Different performance criteria can be used for this, depending on the users' preferences. Now, the design hypotheses can be tested. This is a non-trivial process in which the designer aims to assess the extent to which particular theory components have contributed to the system's performance. Based on this assessment, the validity of the theory components for addressing design problems *of this particular type* will be adjusted and added to the design theory knowledge base.

The idea of theoretical components having a validity requires some elaboration. What we mean by this is that theoretical components (e.g. constructs, models, methods, measures etc.) have associated with them a set of hypotheses. If the number of tested hypotheses indicating a positive correlation between the theory component and IS performance increases, then we say the overall validity of the theory component increases. If there are no or conflicting test results, the validity decreases. For example, if time and again it turns out that using a technical facilitator to support users in their use of complex electronic meeting room software increases performance dramatically, then the continuously confirmed hypothesis "using a facilitator increases the effectiveness of meeting room software" increases the validity of the theoretical component "*heuristic*: technical facilitators should be used to facilitate electronic meeting room discussions". Note that many theoretical components will be in the form of hypotheses themselves (e.g. heuristics), although not necessarily so. For instance, techniques are also theoretical

[2] Hevner et al. (2004) distinguish between 'foundations' (including theories) and 'methodologies' as main categories in the Knowledge Base. For our purposes, this definition is rather artificial, and we consider all of them 'theoretical components', although they can be more conceptual or more methdological in nature.

components, but not in hypothetical form. However, about whether this technnique solves (or not) particular design problems, many hypotheses can be formulated and tested.

2.2 Community IS Design

Community IS are a special class of information systems, both in terms of the artifacts and the methodologies used. Interesting community-specific theoretical components abound. An example of a conceptual component is the concept of sociability (Preece, 2000). Examples of a methodological component are numerous design heuristics, e.g. on the role of facilitation in building online communities (Preece, 2000; Wershler-Henry & Surman, 2001). These theoretical components are necessary and useful for design purposes. Still, which components to use for which types of community design problems, and what interaction effects occur when combining theoretical components, is largely unknown.

Progress towards a more coherent, deep theory is still very much lacking.in the area of community IS design. A major cause of the lack of conceptual progress, which applies to the IS field in general, and the Community IS research community in particular, is the fragmentation of research efforts. (Unhealthy) fragmentation is not equal to (healthy) pluralism. When pluralism prevails in a research community there is a diversity of ideas, perspectives, research approaches and paradigms, but a shared underlying core set of knowledge or beliefs. In case of fragmentation, there is insufficient communication between different (sub)communities, and no such – necessary – core knowledge set exists (Hirschheim and Klein, 2003).

There are several reasons for this state of fragmentation in community IS design research. The field is still very young, even younger than the IS field in general. Furthermore, because of cultural differences, there is large gap between theorists and practitioners in this domain. Often rightly so, practitioners working with and in communities have a sense of hostility towards the many researchers still presenting quick technological fixes for very complex social problems. A third, more fundamental problem, has to do with the nature of community IS, which, of all types of IS development probably most requires a systemic and longitudinal approach.

Much research on online communities so far has focused on studying technological functionalities; the linguistic and (inter)action behaviors of community members and the social networks in which they operate; and the drivers for cooperation and participation in virtual communities (e.g. Schubert & Koch; 2003; Huysman et al., 2003; Cassell & Tversky; 2005; Kozinets, 1999). However, true community information systems development requires a careful study of the continuous process of co-evolving complex social and technical systems. New tools lead to new practices and ways of working, which in turn lead to new affordances for and constraints on technical innovation (Winograd, 1995). Communities cannot be declared, but need to be slowly grown over a long period of time (Wenger et al., 2002). Instead of studying isolated relations between, for example, a particular design intervention and an increase of effectiveness in a community, they should be examined as embedded in complex socio-technical systems, characterized by interdependencies and long-term evolution. One way to conceptualize such systems, particularly applicable to communities with their fuzzy, dynamic, and permeable nature, is as "socio-technical networks". These are interrelated and interdependent milieus of people, their social and work practices, the norms of use, hardware and software, the support systems that aid users, and the maintenance systems that keep their ICTs operating (Lamb et al., 2000). To design such networks, traditional waterfall-based systems development approaches, with their clear stages, deliverables and well-understood dependencies no longer suffice (Brooks, 1995). Instead, more holistic views are required, particularly in the design stage, where perceived effects of community IS interventions often turn out to be quite different from intended effects due to the socio-technical network complexities.

The challenge is to go beyond reactively studying information systems as change agents and instead to pro-actively improve specific ways of engineering systems that can contribute to desired changes in the environment (Purao & Truex, 2004). With respect to community IS this is even more important, since interventions are so hard to design, because of the systemic, longitudinal and situated nature of communities. How then to arrive at better, more coherent design theories for this very complex domain? Simulation, more in particular, system dynamics theory, could provide a way out.

3 System Dynamics: Theory Meets Practice

Much systems behaviour is counterintuitive. Many people are not able to visualize the exponential, non-linear effects of interventions in complex systems. We extrapolate linearly, but much real-world behaviour is much more complex, because of dependencies between variables and feedback loops in which the output of a system component ultimately has an effect as an input in the future. Additional complexity is introduced by the existence of accumulations (stocks) and delays. People often dramatically underestimate the inertia of systems, leading to incorrect decisions with a short term focus. To handle such complexity, computer-aided simulations are indispensable. System dynamics is a methodology particularly suited to analyze such complex, large-scale, non-linear, partially qualitative, dynamic systems (Sterman, 2000). System dynamic models can be used to generate and analyze very complex, realistic behaviour. However, they consist of combinations of only a few simple conceptual building blocks, the most important ones being stocks, flows, feedback loops, and delays.

Stocks are accumulated quantities or resources, characterizing the state of the system. Stocks give systems inertia and memory. A stock continues to exist, even if all the dynamics of the system come to a halt. An example of a stock is the number of community members at a certain point in time.

A *flow* is a change to a stock that occurs during a period of time. A flow that is an input to a stock is called an inflow, a flow that departs from a stock is an outflow. A stock can only grow or deplete by its inflows or outflows. An example of an inflow of the stock is the average number of new members joining a community per unit of time.

Feedback loops are the backbone of system dynamics models. All systems consist of networks of positive and negative feedback loops. The resulting dynamics arise from the interaction between these loops and can result in very complex behaviour. *Positive feedback loops* reinforce what is happening in the system. An example of such a loop would be: more investment in community facilitation leads to better quality discussions, which leads to a higher external reputation, which leads to more investment in community facilitation. A *negative feedback loop* counteracts change, and is self-correcting, in the sense that it stabilizes around a certain parameter value. For example, a higher external reputation leads to more commercial interest, which leads to less community spirit, which leads to a lower external reputation.

A *delay* is a process whose output lags behind its input in some fashion, and is modelled by stocks and flows. For example, say there is a stock 'community members-in-training'. A delay of one month represents that community members 'stay' in that stock for that period of time before 'flowing' to the next stock of 'trained community members'.

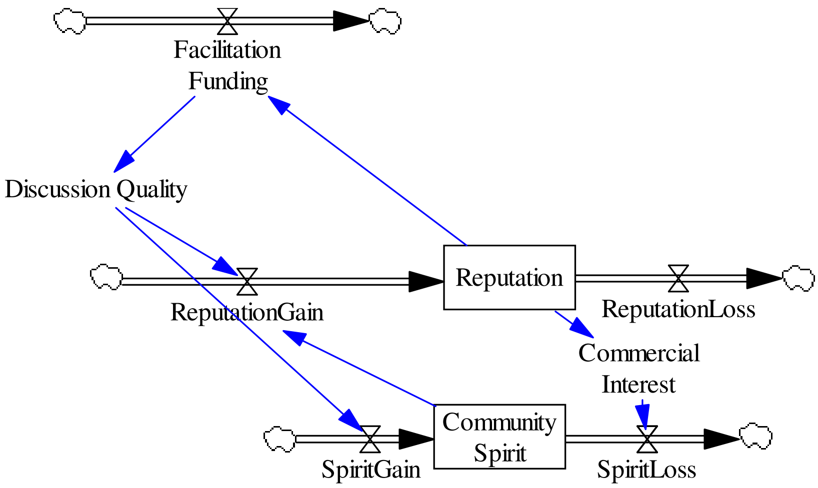

Figure 2 A System Dynamics Model of a Virtual Commmunity

To illustrate the essence of system dynamics, we show a model of a virtual community containing the two feedback loops described above (Fig.2). There are many ways to operationalize such constructs and loops. There is not one, best, model. In fact, one could even say that "all models are wrong" in that they should not be used to accurately predict values of individual variables. However, they can be very useful in obtaining a general understanding of the overall behaviour of a system, and to isolate the effects of single variables *in a system context* (Sterman, 2002; Campbell, 2000).

One key concept of interest in our model is that of ‘community spirit’, which we have modeled as a stock. The level of community spirit is positively related to reputation gain. This means that if community spirit rises, reputation is gained, if it is lowered, reputation gain decreases. Community spirit itself is gained by a higher discussion quality, spirit is lost if the community attracts too much commercial interest. To operationalize this model, we chose some plausible values for stocks, rates, and (auxiliary) variables. In Fig. 3, we show an example of output of this model for this choice of values. It shows very well the complex behaviour generated by different loops being combined. Even though this model is still very simple, behaviour already is very difficult to assess with the unaided human mind. System dynamics offers a full array of tools to perform sensitivity analyses and other types of techniques to help better understand causes and effects in real world systems full of interdependencies and feedback.

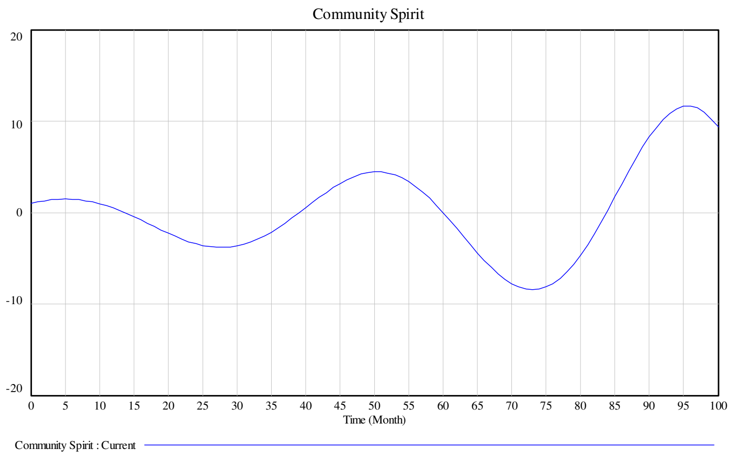

Figure 3 The output for the stock 'Community Spirit''

It is not within the scope of this paper to discuss system dynamics in detail. There is a vast literature, set of methodologies, and tools on this subject. A good primer is given in (Sterman, 2000). Our purpose here is to clarify the role that system dynamics could play in developing better theories, or rather, better developing theories, on community IS design.

4 Using System Dynamics to Improve Theory Development on Community IS Design

In the past decades, system dynamics has been widely applied to help analyze problems in many domains, ranging from the global environmental crisis to business and logistical problems. Virtual community researchers as well have started to take notice of this interesting set of methodologies and techniques. For instance, Diker (2004) has created an impressive model of growth policies in open online collaboration communities. However, it is our claim that besides being useful to better understand the *operational* behaviour of virtual communities, system dynamics could also play a crucial role at the meta-level of community IS design *theory development*. In this section, we will sketch one way in which this could be implemented.

Community information systems are complex, evolving socio-technical systems, which require a *situated*, *longitudinal* and *systemic* design approach. This means that each community has unique IS design needs, that the effects of design interventions need to be studied in the long term to become fully visible, and that the effects of interventions (such as using a particular technique to solve a particular design problem) can only be studied in their interactions with many system elements and context factors.

The situated nature of community information systems implies that any *design solution* (in terms of particular selections and combinations of theoretical components) may be only relevant to that particular design problem *instance*. The generalization of a design solution applying to a *class* of instances will always be difficult. Statistical generalization, in the sense of making inferences about a population on the basis of empirical data collected about a sample is generally not possible. Analytical generalization, however, where an investigator tries to identify connections between the findings of the case to a broader theory based on intelligent reasoning, is a valid approach, (Yin, 1994). System dynamics can help in such analytical generalization, by clarifying consequences of design choices based on the theoretical commitments one makes. The following method, building on the meta-model of IS development we presented earlier, shows one way to go forward (Fig.4):

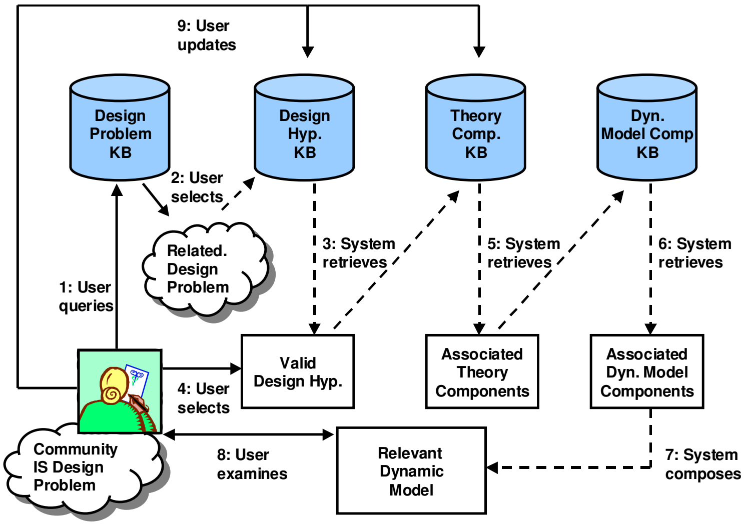

Figure 4 Community IS Design Theory Development Using System Dynamics

The procedure could be as follows, here presented as an interaction between a human user and a supporting Theory Development System:

- To aid an investigator in theory development, three primary knowledge bases are needed: (1) a knowledge base of Design Problems, (2) a knowledge base of Theory Components, and (3) a knowledge base of Design Hypotheses.
  - The *Design Problem* knowledge base contains cases, examples, reports and so on describing design problems, including relevabt problems related to the use, development, and maintenance of community IS. The problem descriptions can be in the form of case studies, project report excerpts, hypothetical scenarios, etc.
  - The *Theory Component* knowledge base has pointers to a whole array of theoretical components. Hevner et al (2004), in their Knowledge Base list the following quite complete list of 'Foundations' and 'Methodologies': theories, frameworks, instruments, constructs, models, methods, instantiations, data analysis techniques, formalisms, measures, and validation criteria.
  - The *Design Hypothesis* knowledge base contains information on how well a particular theory component or combination of components has addressed (based on implementation and use experiences) a particular design problem. Validity can be indicated in different ways: source of the data, reviews, conditions of the experiment, links to other theory components, and so on.
- The Theory Component knowledge base is a source of information for the development of a Dynamic Model Component knowledge base. Whereas the first three knowledge bases need to be filled by researchers and practitioners from the Online Communities domain, the Dynamic Model Component knowledge base needs to be filled by system dynamics experts. Each theory component can have one or more dynamic model components associated with it.
- Now if a researcher encounters a design problem for which no solution has yet been found, she can query the Design Problem knowledge base, and identify the problem that best seems to fit her case. The system presents her with hypotheses (retrieved from the Design Hypothesis knowledge base) that have proven to be successful in other cases dealing with the selected design problem. The user selects the hypotheses she finds most valid, according to her own criteria. The system then first retrieves the theory components

associated with the selected hypotheses. Subsequently, it retrieves the dynamic model components that are associated with these theory components. It then automatically composes an overall dynamic model out of these components (assuming that any inconsistencies between these models have been dealt with).

- The researcher can now perform a range of simulations, which will give her a much deeper understanding of the design problem, solutions, and their interrelationships. Based on the results, she can decide that the selected problem or design hypotheses were not right, or that theory components or design hypotheses need to be modified or created. The process can then be iterated as needed.

This conceptual outline is not necessarily to be taken literally. It is a scenario that could be implemented at various levels of scope and detail. From the point of view of an individual researcher, it might mean that she makes a classification of the kind of problems she is working on, the theoretical components she uses, and what here core design hypotheses are. She could then build some dynamic models and start working systematically at furthering her personal understanding. On the other hand, a much more ambitious, yet not necessarily infeasible endeavor would be to have an international network of community researchers work jointly on the realization of this vision. A natural candidate for this would be the Community Informatics Research Network[3], which has as its mission to promote and represent community informatics and community networking research internationally.

5 Conclusions

This paper is titled "Towards a design theory for community information systems". Our purpose was not to propose the ultimate, or even a tentative design theory. Instead, we focused on how the *process* of getting to such a theory can be understood, charted and facilitated. A better understanding of the design theory development process is a first step, as exemplified by our metamodel of the IS development process. However, a second step is also needed. The complexity of design theory formation in the domain of community IS is very high. Support by the analytical machinery of systems dynamics is a powerful way to build a more integrated core of conceptual knowledge in our domain. Theory formation benefits from a strong theoretical-empirical cycle. In other fields, such as the natural sciences, experimentation is common to get to such an integrated body of paradigmatic knowledge. In information systems research, particularly in community IS design, this is very difficult to achieve however, because of the situated, longitudinal and systemic aspects of design research. Enough experiments of adequate complexity simply cannot be carried out to detect what are the invariant patterns that could be the basis for solid theory formation. Simulation through system dynamics could therefore become an important research catalyst for our field.

We gave only a toy example of the use of simulation and merely a sketch of a method for theory development using system dynamics. Many issues need to be addressed before the outline presented here can be realized. For example, to obtain a true body of deep knowledge, better and many more community IS dynamic model components need to be developed, based on solid theoretical and practical findings. Interfaces between such model components need to be clearly defined. This is not trivial, since there are many epistemological differences between the theory components on which they should be based. Many interesting empirical findings already exist, but they are scattered across practice and literature. Such data need to be translated into indicators and parameter values of the dynamic models to which they apply. Another issue is how to measure the utility of particular IS solutions for particular design problems? How to use these measurements in our SD models? System dynamics has developed a rich set of tools and techniques to assist in the analysis and understanding of very complex dynamic phenomena. Which ones would be most suitable for the typical research problems community IS researchers face? How can simulation results best be used in validating theory components?

This paper only addressed some of the many (meta)theoretical issues related to community IS design theory formation. We also only gave a brief sketch of how system dynamics could be practically embedded in the research process of our community. Still, we hope the ideas presented here give enough food for thought to trigger a useful discussion about developing a feasible and much needed more systematic approach to do more fundamental community IS research.

References

---

[3] http://www.ciresearch.net/

Andrews, D. (2002). Audience-Specific Online Community Design. *Communications of the ACM*, 45(4), 64-68.

Brooks, F. P. (1995). The Mythical Man-Month : Essays on Software Engineering (anniversary ed.). Reading, MA: Addison-Wesley.

Campbell, B. (2000). System Dynamics in Information Systems Analysis: An Evaluation Case Study. In D. Bustard, P. Kawalek & M. Norris (Eds.), *Systems Modeling for Business Process Improvement* (pp.33-46), Norwood, MA: Artech House.

Cassell, J. & Tversky, D. (2005). The Language of Online Intercultural Community Formation. *Journal of Computer-Mediated Communication*, 10(2).

Diker, V. G. (2004). A Dynamic Feedback Framework for Studying Growth Policies in Open Online Collaboration Communities. In *Proc. of the Tenth Americas Conference on Information Systems, New York, August 2004*.

Gongla, P. & Rizzuto, C. R. (2001). Evolving Communities of Practice: IBM Global Services Experience. *IBM Systems Journal,* 40(4), 842-862.

Gurstein, M. (2003). Effective Use: a Community Informatics Strategy Beyond the Digital Divide. *First Monday,* 8(12).

Hevner, A. R., March, S. T., Park, J. & Ram, S. (2004). Design Science in Information Systems Research. *MISQ,* 28(1), 75-105.

Hirschheim, R. & Klein, H. K. (2003). Crisis in the IS Field? A Critical Reflection on the State of the Discipline. *Journal of the Association for Information Systems*, (4)5, 237-293.

Hirschheim, R., Klein, H. & Lyytinen, K. (1995). Information Systems Development and Data Modeling: Conceptual and Philosophical Foundations. Cambridge University Press.

Huysman, M., Wenger, E. & Wulf, V. (Eds.).: 2003, Communities and Technologies. Dordrecht: Kluwer Academic Publishers.

Kozinets, R. V. (1999). E-Tribalized Marketing? The Strategic Implications of Virtual Communities of Consumption. *European Management Journal* 17(3), 252-264.

Lamb, R., Sawyer, S. & Kling, R. (2000). A Social Informatics Perspective on Socio-Technical Networks. In H. M. Chung (Ed.), *Proc. of Americas Conference on Information Systems, Long Beach, CA*.

Markus, M. L., Majchrzak, A. & Gasser, L. (2002). A Design Theory for Systems that Support Emergent Knowledge Processes, *MISQ,*, 26(3), 199-232.

Preece, J. (2000). Online Communities : Designing Usability, Supporting Sociability. Chichester, NY: John Wiley.

Purao, S. & Truex III, D. P. (2004). Supporting Engineering of Information Systems in Emergent Organizations. In B. Kaplan, D. P. Truex III, D. Wastell, A. T. Wood-Harper and J. I. deGross (Eds.), *Information Systems Research: Relevant Theories and Informed Practice.* Boston: Kluwer Academic Press.

Schubert, P. & Koch, M. (2003). Collaboration Platforms for Virtual Student Communities. In *Proc. of the 36th Hawaii International Conference on System Sciences (HICSS-36), Hawaii, January 2003*. IEEE Computer Society.

Schubert, P. and Ginsburg, M. (2000). Virtual Communities of Transaction: The Role of Personalization in Electronic Commerce. *Electronic Markets*, 10(1), 45-55.

Sterman, J. D. (2000). Business Dynamics: Systems Thinking and Modeling for a Complex World. London: Irwin McGraw-Hill.

Sterman, J. D. (2002). All Models Are Wrong: Reflections on Becoming a Systems Scientist, *System Dynamics Review*, 18(4), 501-531.

Wershler-Henry, D. & Surman, M. (2001). Commonspace: Beyond Virtual Community. Toronto: Pearson.

Walls, J., Widmeyer, G. R., & El Sawy, O. A. (1992). Building an Information System Design Theory for Vigilant EIS, *Information Systems Research,* 3(1), 36-59.

Wenger, E., McDermott, R. & Snyder, W. (2002). Cultivating Communities of Practice. Cambridge, MA: Harvard Business School Press.

Winograd, T. (1995) From Programming Environments to Environments for Designing. *Communications of the ACM*, 38(6), 65-74.

Yeates, D., Shields, M., & Helmy, D. (1994). Systems Analysis and Design. London: Pitman.

Yin, R. (1994). Case Study Research: Design and Methods (2nd ed.). London: Sage.