

# Using System Dynamics to Construct Design Theory on Community Information Systems

Aldo de Moor

CommunitySense, the Netherlands

[ademoor@communitysense.nl](mailto:ademoor@communitysense.nl)

## Abstract

*Virtual communities are complex and evolving socio-technical systems. Community information systems require much theoretical research to address concrete design problems. Community IS design theory is still very young and fragmented, however. In this article, we outline a design theory construction methodology that could help build conceptual linkages between the disparate collection of (proto)-theories, heuristics, and empirical data available in community informatics. The aim of such a methodology is to more efficiently define, select, link, and evaluate theoretical insights about which technologies and procedures to use to satisfy particular (community) information or communication requirements. We explore how simulation via system dynamics could play an important role in the more systematic development of design theory for community information systems.*

## 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. Andrews, 2002). Other research examines the role of specific technologies in 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 access to or use of technologies, such as problems with respect to the Digital Divide and effective use (e.g. Gurstein, 2003).

Most of this research focuses on elements or aspects of the technologies supporting communities. Another important line of research, concerns the *process* in which the information systems of technologies supporting communities can be developed. Such systems are not identical to the virtual communities themselves, but rather form the technical support systems that support (human) communities in their information and communication needs. In the development process of such systems, a mix of technologies is tailored to the unique needs of a particular online community. Some well-known approaches addressing this problem are Wenger et al.'s (2002) approach to cultivating communities of practice, Gongla & Rizzuto's approach to evolving communities of practice (2001), and Preece's (2000) method for community-centred development. Still, systematic methods for the development of community information systems solidly grounded in theory are rare and may as yet be premature.

IS research is still a young discipline. Paradigms, concepts, models, methods and techniques are in the early stages of development, compared to the more established natural and social sciences. Much Information Systems (IS) development theory is inspired by a behavioral science paradigm, in which the goal is to develop and justify theories that explain or predict the organizational and human phenomena surrounding information systems analysis, design, implementation, management and use. A complementary perspective 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 study and development of information systems can be accomplished effectively and efficiently (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 say 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, 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 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. 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). Community IS development also has specific 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).

The aim of this paper is to help provide the community informatics-community with the methodological tools to start working towards design theories specific for community IS. In this paper, we will *not* address the content of such theories: the specific characteristics of community IS development and the approach to turning requirements into effective systems demands the collaboration of many theorists and practitioners in the field. Instead, we outline a methodology that could help build conceptual linkages between the disparate collection of (proto)-theories, heuristics, and empirical data available in community informatics. The aim of such a methodology is to more efficiently define, select, link, and evaluate theoretical insights about which technologies and procedures to use to satisfy particular (community) information or communication requirements. A starting point for such a methodology is current work in IS design theories. The methodology outlined revolves around a workflow for selecting a design problem, relevant design hypotheses, theory components and dynamic model components. We use system dynamics as an approach to visualizing and examining the implications of the selected theoretical components.

In Sect.2, we explore the characteristics of community IS design. Sect. 3 examines the notion of IS design theories which leads to our meta-model of IS design theory development. In Sect. 4, we introduce system dynamics and outline our methodology for using it in community IS design theory construction. Sect. 5 and Sect. 6 contain a discussion and conclusion. Appendix 1 describes in detail a hypothetical scenario about constructing theory on conflict resolution in online gaming communities.

## 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, comprising both conceptual and methodological components, abound. An example of a conceptual component is the concept of sociability (Preece, 2000). Examples of methodological components 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 the design purposes of the information systems needed by communities<sup>1</sup>. 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 there is also a shared underlying core set of knowledge or beliefs. In the 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 express 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. *Systemic* means that the many different components (tools, procedures, functionalities, roles, persons...) can only be appropriately analyzed, designed, implemented, and evaluated as part of the information system-in-its-context-of-community-use. *Longitudinal* refers to such a system evolving over a considerable period of time, since it is not possible to declare a community, only to grow it with much trial-and-error.

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

---

<sup>1</sup> *The focus in this article is on online or virtual communities, since their communication and information processes are most clearly mediated by information technologies. However, there is no clear boundary between virtual and physical or geographical communities, such as neighbourhood communities. Most geographical communities nowadays also make use of the Internet for coordinating their physical activities. Vice versa, most online communities after some time organize events where members can meet face-to-face. In this article, the focus is on communities at least partially supported by information technology (and requiring design of effective information systems). The exact terminology to classify such communities and their systems is of less interest to us.*

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 also 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 of 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). Studying isolated relations between, for example, a particular design intervention and an increase of effectiveness in a community, is not of much use. Instead, these relations 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 milieux of people, their social and work practices, hardware and software, the norms of use, 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 the realized effects of community IS interventions often turn out to be quite different from the 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 due to the systemic, longitudinal and situated nature of communities. How then to arrive at better, more coherent design theories for this very complex domain? Before we can suggest one possible answer this question, we first need to take a closer look at how these theories can be understood.

## IS Design Theories

As noted in the previous section, community IS are the systems of interacting technologies supporting the information and communication requirements of human communities. Such systems have special properties in terms of content and functionalities, compared to more traditional information systems such as banking or payroll systems that typically handle (transaction) data and information.. Still, many of the *meta*-design theoretical principles related to the *development* of such systems are useful, at least as a starting point.

Information systems 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 sub-divisions within the systems development process. One widely accepted interpretation of two of its key stages is the following: *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 results of the analysis (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).

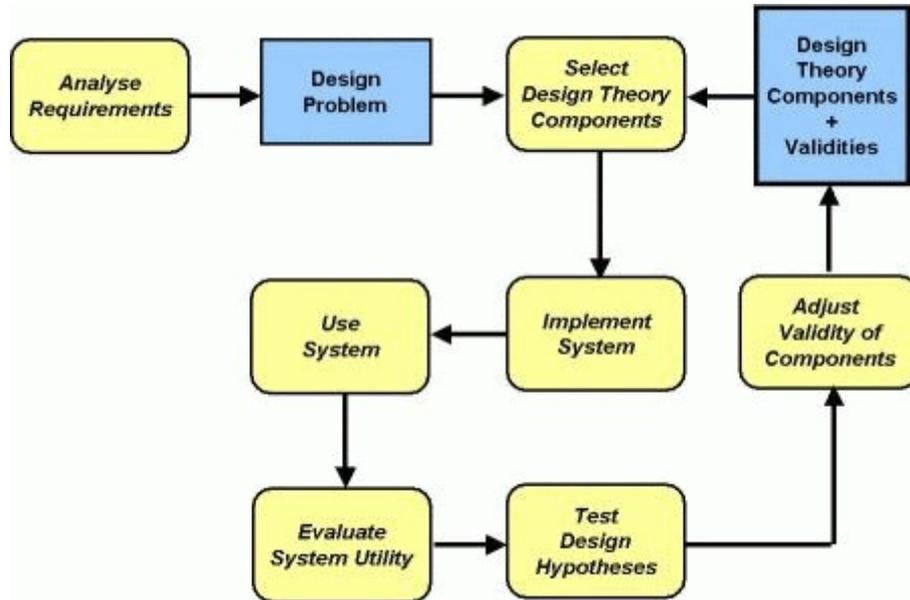
The purpose of a *design theory* is to support the achievement of goals, contrary to the purpose of *natural science theory*. Walls et al (1992) give a good introduction to design theories for information systems. They show how explanatory, predictive, or normative theories can be put to practical use. 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 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 an 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 must therefore 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).

## 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 to capture the development of a particular (community or traditional) IS, but rather the *process* of developing design theories for IS. Such a systematic approach is important for more efficiently defining, selecting, linking, and evaluating insights about which technologies and procedures to use to solve particular (community) information or communication requirements.



**Figure 1 A meta-model of IS development**

At the heart of our IS design theory development meta-model are *theoretical components*<sup>2</sup>, 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) in helping to solve the problem. To capture the rationale for this decision, the designer formulates one or more design hypotheses which state how the theory component could contribute to the resolution of the problem<sup>3</sup>. 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.

The design hypotheses can now 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 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 and weight of the 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.

<sup>2</sup> 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 methodological in nature.

<sup>3</sup> An important source of such design hypotheses could be the design patterns for "living communication", collected by Doug Schuler in the Liberating Voices! project: <http://trout.cpsr.org/program/sphere/patterns/>

## Using System Dynamics for Community IS Design Theory Development

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. 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.

In this section, we first present a short introduction to system dynamics, then show that it can be used as a fundamental element of a framework for better community IS design theory development.

### 1. System Dynamics: Theory Meets Practice

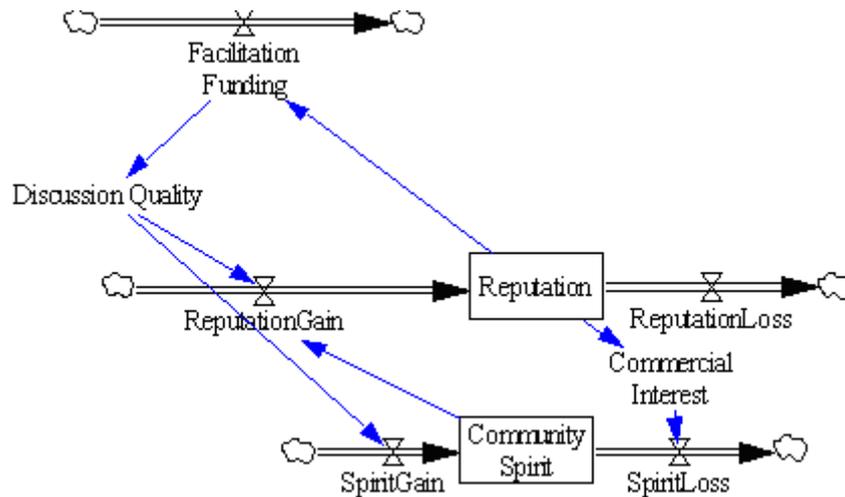
Much systems behaviour is counterintuitive. Many people are not able to visualize the exponential, non-linear effects of interventions in complex systems. We extrapolate linearly, but much real-world behaviour is rather more complex, because of dependencies between variables and feedback loops in which the output of a system component ultimately has an effect as a future input. 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 behaviours. *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.

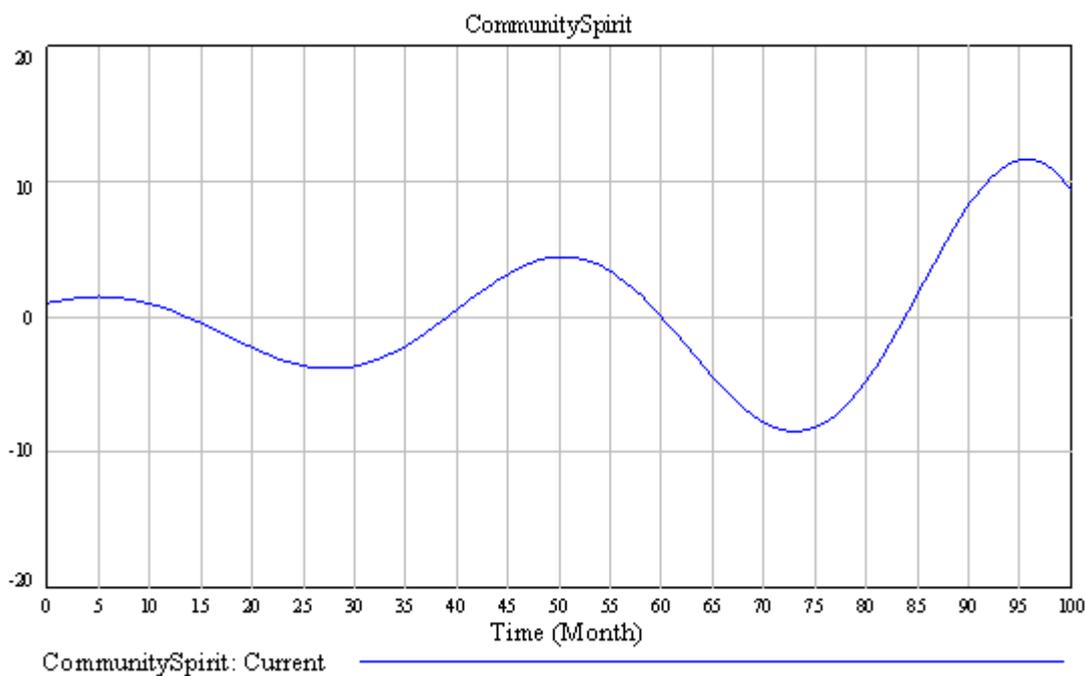
A *delay* is a process whose output lags behind its input in some fashion, and is modelled by stocks and flows. For example, say there is a stock ‘community members-in-training’. A delay of one month represents that community members ‘stay’ in that stock for that period of time before ‘flowing’ to the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



**Figure 2** A system dynamics model of a virtual community

To illustrate the essence of system dynamics, we show a model of a virtual community containing the two feedback loops described above (Fig.2). One core concept in this 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.

There are of course 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 (Sterman, 2002). 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* (Campbell, 2000). 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.



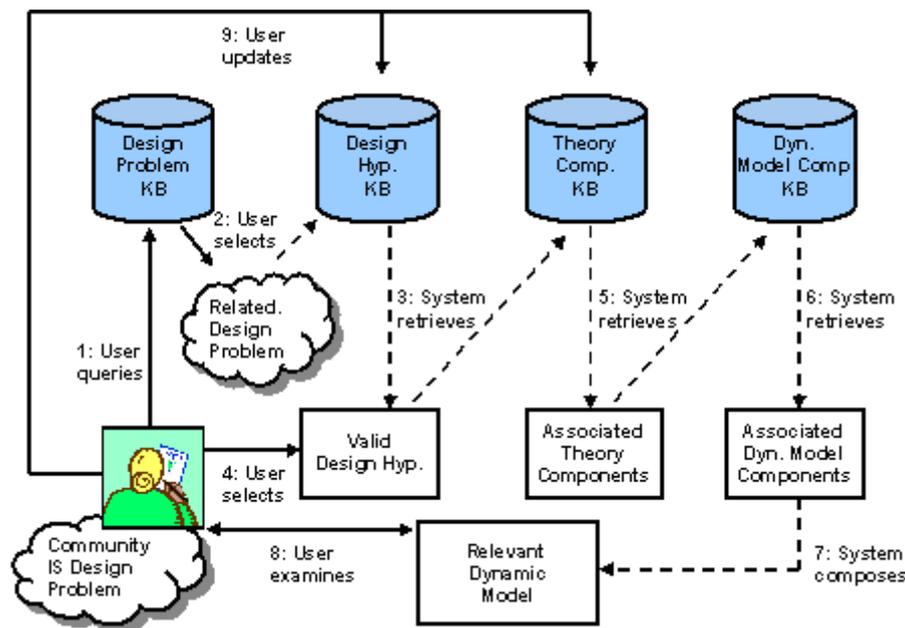
**Figure 3** The output for the stock 'Community Spirit

It is not within the scope of this paper to discuss system dynamics in detail. There is a vast literature, and many

methodologies, and tools related to this subject. A good primer is given in (Serman, 2000). Our purpose here is to clarify the role that system dynamics could play in developing better theories, or rather, better developing theories, towards community IS design.

## 2. Towards a System-Dynamics Based Methodology for Community IS Design Theory Construction

Here, we outline a methodology for community IS design theory construction, combining the meta-model of IS development presented earlier, along with system dynamics as the components of the dynamic model (Fig.4):



**Figure 4 Community IS design theory development using system dynamics**

The steps of the methodology, here presented as an interaction between a human user and a supporting Design Theory Development System, are as follows:

- To aid an investigator in theory development, four primary knowledge bases are needed: (1) a knowledge base of Design Problems, (2) a knowledge base of Design Hypotheses, (3) a knowledge base of Theory Components, and (4) a knowledge base of Dynamic Model Components<sup>4</sup>:
  - The *Design Problem* knowledge base contains cases, examples, reports and so on describing design problems, including 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, and so on.
  - 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 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 *Theory Component* knowledge base is a source of information for the development of a *Dynamic Model Component* knowledge base. This knowledge base contains operationalizations of the theories described in the *Theory Component* knowledge base. Such operationalizations, including labels for concepts and values for parameters, are typically obtained in empirical studies testing one or more design hypotheses.
- If a researcher encounters a design problem for which no solution has yet been found, she can query the Design

<sup>4</sup> Ideally, the construction of these knowledge bases should be done by the at large community of community informatics researchers and practitioners’ over an extended period of time.

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<sup>5</sup>. 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 helps the researcher to compose an integrated 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 the 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 an idea that could be implemented in various ways and levels of 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 her core design hypotheses are. She could then build some dynamic models and start working systematically at furthering her personal understanding. A much more ambitious, yet not 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<sup>6</sup>, which has as its mission to promote and represent community informatics and community networking research internationally. To give a more concrete idea of the many possible uses of simulation in community informatics theory construction and validation, Appendix 1 presents one hypothetical scenario for constructing theory on conflict resolution in gaming communities.

## Discussion

What happens in simulations? In an iterative process, a researcher efficiently constructs her own proto-theory using and combining design hypotheses from previous theories that have already been tested to a greater or lesser extent. System dynamics simulations helps her to combine assumptions from different theories and get a sense of the impact over time of particular design hypotheses on the overall behavior of the community. It is thus possible for her to get a much better sense of possible interactions between theoretical assumptions than would be possible when just looking at static relations in models from different sources. This, in turn, gives her a good idea of what are interesting relationships to test.

Of course, the results of simulations should not be taken at face value. They are not descriptions of reality, nor precise prescriptions of the value of particular variables. Unfounded relations and parameter choices will result in nonsensical simulations. Moreover, even if the relations and parameters have been properly empirically validated, the simulations can still produce useless outcomes, because relations or parameters crucial to the case at hand have not been taken into account in the model.

Despite these limitations, the value of system dynamics models is in (mercilessly) showing the consequences of conceptualizations, assumptions, and design choices made by the researcher. Even though the actual values should not be taken too seriously, the overall trends and directions represent educated guesses of dependencies and determinants of community behavior. In a nascent field like community informatics, which is growing from such a hodgepodge of paradigms, models, and empirical findings, system dynamics models thus can make a useful contribution as integrating tools for testing the coherence, consistency, and relatedness of proto-theories. In this instance, such a simulation approach would need to be used in theory-constructing, not testing, as it is only useful in extrapolating the consequences of one's theoretical assumptions.

System dynamics can be a useful complement to methodologies such as *grounded theory*. There, concepts and relationships describing and "owned by" a community are being extracted by means of a formal methodology (Urquhart, 2001; Fernández and Lehmann, 2005). An approach as described in this paper can help to integrate such "local theories" in more widely accepted, generic theoretical frameworks. System dynamics models have also proven themselves very useful as a tool to elicit knowledge from experts. In interview sessions, based on simulating different scenarios, high-quality observations by experts on parameters, concepts, and relations can be obtained.

Truly understanding "meta-aspects" of design theory construction and being familiar enough with methodologies like system dynamics, requires considerable practice. A researcher needs much experience to deeply understand the meaning and implications of the choices proposed here. However, this is no different for any other professional methodology. Furthermore, the examples given in this paper are only meant to show proof of concept, not to be used directly as-is. Much of the complexity can be hidden by using better defined processes and interfaces. It will be a major

---

<sup>5</sup> Such matches between problem and hypothesis, hypothesis and theoretical component, and theoretical component and dynamic model component need in practice, to be nothing more than a manual link added by a researcher or practitioner who thinks the link to be of value.

<sup>6</sup> <http://www.ciresearch.net/>

R&D challenge, but doing so could make a significant contribution to improving the quantity and quality of design theories on community IS.

## Conclusions

Our purpose in this paper 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 could be understood, charted and facilitated. A better understanding of the design theory development process is a first step towards this objective, as exemplified by our meta-model. However, a second step is also needed. The complexity of design theory formulation 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 formulation 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 formulation.

Simulation through system dynamics could therefore become an important research catalyst for our field. System dynamics is not a panacea, but a useful additional instrument for reflection and integration of disparate pieces of the community IS design theory development puzzle. It helps to determine the consequences of theoretical choices, and grounding these in other, already more accepted theories and empirical findings. While starting with a few, high-level relationships and very primitive assumptions, over time more detailed and tested relations and values can be explored, making the models increasingly robust and grounded in reality. Such an “emergent approach” to comparing the effects of different assumptions and findings would be a valuable additional tool for community informatics.

In this paper we have given only a hypothetical example of the use of simulation and merely a sketch of a method for design theory construction 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, many more, and better 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 the literature. Such data needs 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.

Also there is the question of 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? How to systematically link system dynamics to more mature methodologies in community informatics, like grounded theory?

This article addressed only some of the many (meta) theoretical issues related to community IS design theory construction. 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 and more systematic approach to doing community IS design theory.

## Annex 1: A Scenario – Constructing Design Theory on Conflict Resolution in Online Gaming Communities

Jane is a Ph.D. student examining the governance of online gaming communities. She is particularly interested in how large gaming communities handle conflict. She is currently examining a case of one such community, where, despite the use of extensive formal boards and procedures, conflicts are continuously erupting, and are very hard to resolve. She would like to investigate what are the options to improve this situation. As she still needs to define her research questions, she decides to do a Dynamic Model Analysis of her research problem using the *CIRN TheoryMeetsPractice Simulation Portal*. Initially, she wants to focus on those studies that have been successful in dealing with conflict in online gaming communities, hoping that she will come to a better understand the range of interventions possible.

- Selecting a design problem

Jane first accesses the *Design Problem Knowledge Base*. She searches this knowledge base for those studies which say something about both “online gaming” and “conflict”. She gets two hits. One study (Study A) found that in a small-sized online gaming community conflicts increased considerably after a formal conflict management procedure

involving mediation and arbitration was set up. Since the study apparently failed to find successful interventions, she decides not to include this study in her list, at least not initially. The second one seems more promising. Study B concerned a small to medium-sized gaming community, in which conflict was very rare, and if it happened at all, was consistently and easily resolved. Since this case is closer to the size of her own case, and apparently did manage to deal with its conflicts, Jane selects this as her problem description of choice.

- Selecting relevant design hypotheses

From the *Design Hypothesis KB*, the system retrieves the hypotheses that the authors of Study B proposed and tested. One hypothesis of particular interest to Jane is that conflict was reduced because the community members interacted intensively, had known each other personally for a long time, and could access each other's reputation profiles. Thus, the tested (and accepted) hypothesis was that cooperation reduces conflict. A reduction in conflict, in turn, also increased cooperation. Given the quality of the documentation attached to the hypothesis and the international reputation of the authors, the validated hypothesis seems plausible to Jane. She therefore decides to include it in her 'Relevant Design Hypothesis'-selection.

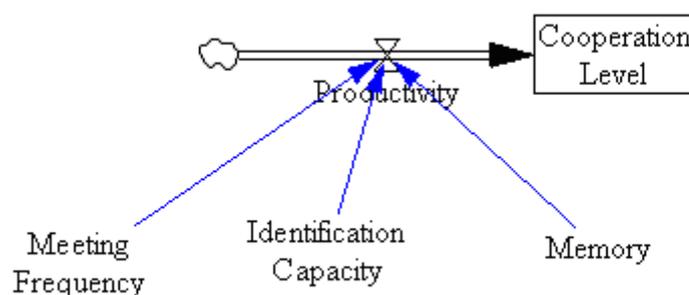
- Selecting associated theory components

Based on her interviews with community members, Jane has the feeling that higher quality cooperation is indeed an important part of the solution to the problem in her community. She decides to further investigate this concept. The authors of Study B based their study on Axelrod's theory of the evolution of cooperation (Axelrod, 1984). Jane does not know this theory, but decides to have a closer look at it. She retrieves the description of Axelrod's theory from the *Theory Component KB* that the authors of Study B used. The description shows the main concepts, their relationships, and whether a relationship is positive or negative with respect to its contribution to "cooperation". The theory claims that there is more cooperation if individuals have met each other more than once, can identify each other, and have information about past behavior. The Theory Component KB also shows all associated design hypotheses in which this theory played a role. It turns out that Axelrod's concept has been used and validated in many hypotheses in the Design Hypothesis KB. This gives Jane the confidence that this conceptualization of cooperation may indeed be a good explanation to explore further. She now knows enough to decide to import this theory component associated with the design hypothesis into her *MyTheoryBase*.

- Selecting associated dynamic model components and examining the full model (iterative)

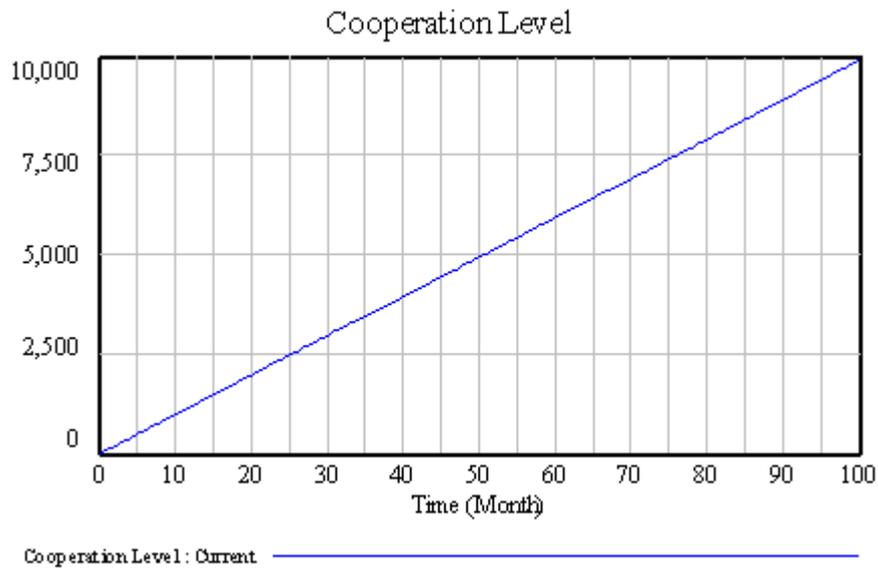
Next, she downloads the operationalization of the theory presented by Study B as stored by its authors in the *Dynamic Model Component KB*. They have operationalized their Axelrod-based concepts as follows: 'Meeting Frequency', 'Identification Capacity' and 'Memory' of past behavior all increase 'Cooperation Output'. The unit of 'Cooperation Level' is an abstract cooperation work unit, initially set at 100. According to Study B, the monthly productivity of the cooperative community is a steady 100 units, to which meeting frequency, identification capacity, and behavior memory all contribute equally. This seems plausible to Jane, not having any further data to the contrary, so she decides to use the Dynamic Model of Study B as the basis for her own operationalization.

The Dynamic Model consists of two linked components. The first one is a simple interpretation of Axelrod's theory (Fig.5).



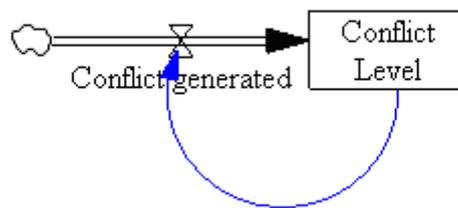
**Figure 5 An operationalization of Axelrod's theory of the evolution of cooperation**

With Productivity = (Behavior Memory+Identification Capacity+Meeting Frequency)\*100; Meeting Frequency = 0.33; Identification Capacity = 0.33; Memory = 0.33; Cooperation Level-initial = 0, the results of the simulation are presented in Fig.6:



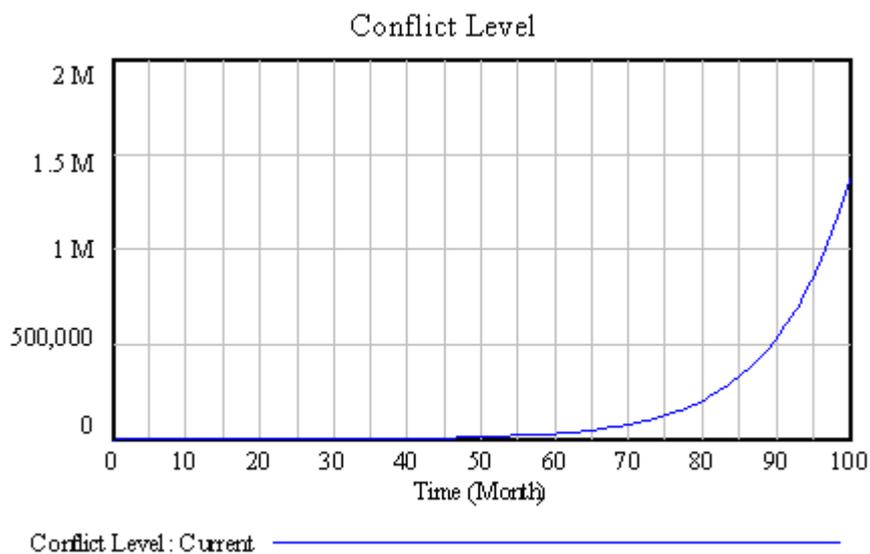
**Figure 6 Results of the simulation of Axelrod's theory**

The second part is a simulation of the conflict level (Fig.7). The authors of Study B have examined a community of 20 people, and found that while initially there is no conflict, that during each period of interaction, a small amount of conflict generation is inevitable (set at 10, again using an abstract unit), and that the amount of new conflict being generated is dependent on the current conflict level. This makes sense, since the more tension there already is in a community, the more tempers will flare and new conflicts will arise ever more quickly.



**Figure 7 An operationalization of conflict generation**

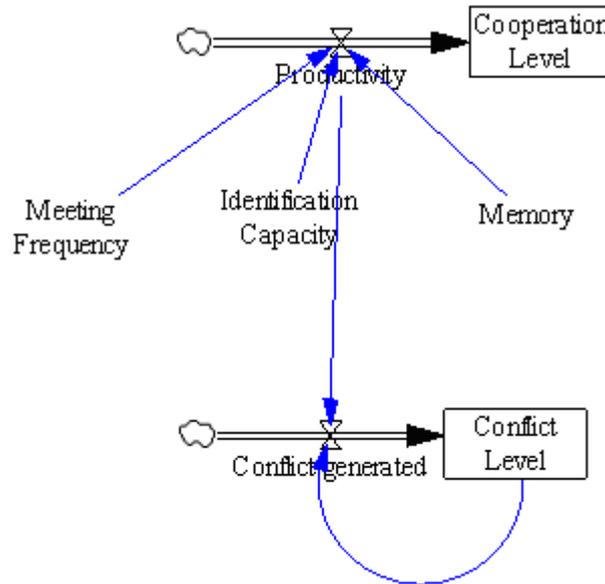
With  $\text{Conflict generated} = 10 + (\text{Conflict Level}/10)$ ;  $\text{Conflict Level-initial} = 0$ , this simulation gives the results of Fig.8:



**Figure 8 Results of the simulation of conflict generation**

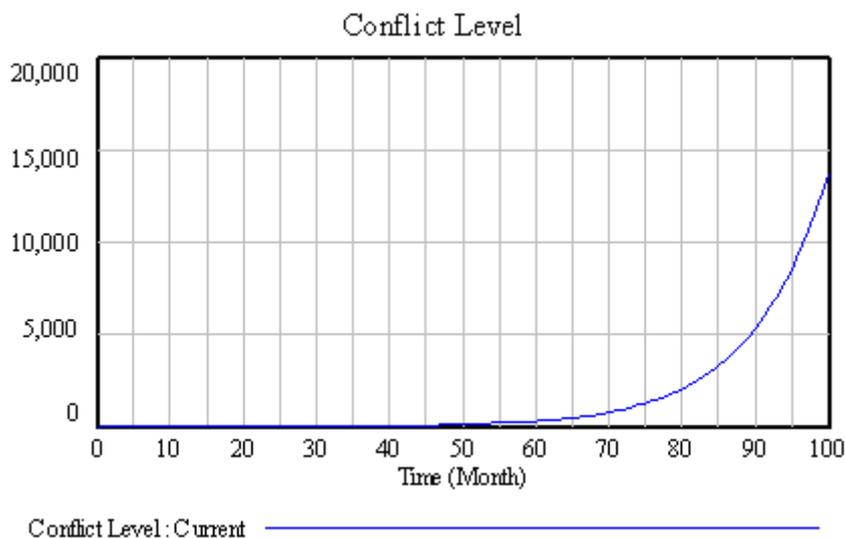
Conflict Level shows a non-linear behavior, very different from the Cooperation Level. This is caused by conflict begetting conflict, at least in this study. In their documentation, the authors warn that over larger time spans, this simulation is unrealistic, if only because a community would have been totally disrupted long before such levels could have been reached in practice. Still, for the limited time spans studied, the model approximated their empirical findings quite realistically.

These two components by themselves are not very interesting. However, when the components are integrated, conflict is quickly reduced because of the dampening effects of cooperation. Thus, Fig. 9 shows the full model underlying the design theory of Study B<sup>7</sup>:



**Figure 9 An operationalization of cooperation-reducing-conflict**

With  $\text{Conflict generated} = 10 + (\text{Conflict Level}/10) - (\text{Productivity}/10)$ ; all other values being the same, the resulting simulation is shown in Fig. 10:



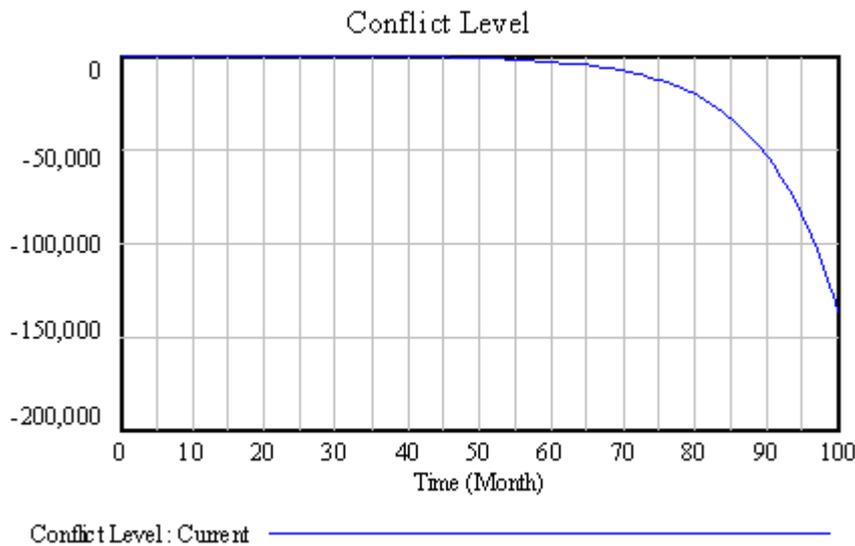
**Figure 10 Results of the simulation of the integrated model**

Although at first sight, the simulation still shows an exponential growth in the Conflict Level, it is much weaker

<sup>7</sup> There are many ways to make the model more realistic, such as Conflict Level feeding back into Cooperation Level. However, the point here is that these dynamic model components expose exactly such shortcomings of theories where the absence of links between variables often remains undetected. Through simulation, one is forced to continuously re-examine one's assumptions and prioritize which of many possible relations between variables to actually model.

than in the previous simulation. Thus, cooperation, at the current rates, does not have sufficient influence to prevent conflict from getting out of control in the end, but keeps it manageable in the short run.

This effect becomes even clearer when giving Productivity only a bit larger weight in conflict generation reduction. For example, with Conflict generated =  $10+(\text{Conflict Level}/10)-(\text{Productivity}/9)$ ; all other values the same (Fig.11):



**Figure 11 Results of the same simulation with increased weight of Productivity**

The resulting simulation even shows a negative level of conflict, in effect meaning that conflict is neutralized completely. Of course, the values used are not to be taken as representing actual empirical data in the collaborative community, but what they do show is that cooperation as conceptualized by the researchers of Study B, can have a strong dampening effect on conflict.

Jane, having seen the Dynamic Model of Study B in action, thinks that it does indeed provide a good starting point for her own research. However, she is not as yet completely satisfied. Whereas Study B’s model shows strong conflict reduction behavior in its community of study, Jane consistently sees conflict getting out of control in her case, even though cooperation levels are high in her community. Obviously, the model needs to be adapted for her situation.

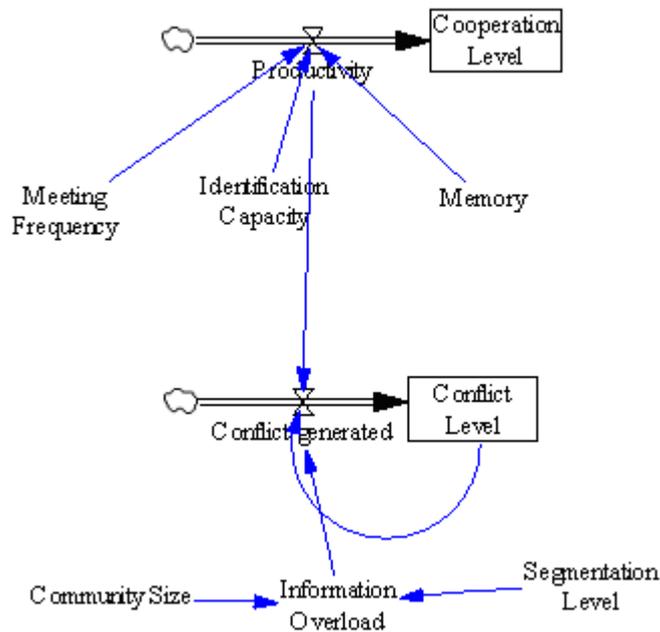
Upon reflecting further, Jane realizes that one key difference between the gaming community described in Study B and her own, is size. Whereas the other community only counted about 20 members, her own has more than 500. From interviews with community moderators, she knows that they often feel overwhelmed by requests for assistance and frustrated that they cannot act promptly on requests for assistance and flames occurring regularly on the community’s discussion fora. She wonders if size may indeed have something to do with her own situation.

She now extends the Dynamical Model of Study B by adding the Community Size factor to it, feeding into the ‘Conflict generated’ rate. As a rough first approximation, she takes the previous equation, and, assuming a linear effect of size increase on conflict generation, multiplies it with the Community Size factor. She takes the value of community size in Study B (n=20) as the original benchmark situation in which size was not taken into account, leading to the following equation:

$$\text{Conflict generated} = (10+(\text{Conflict Level}/10)-(\text{Productivity}/9)) * (\text{Community Size}/20)$$

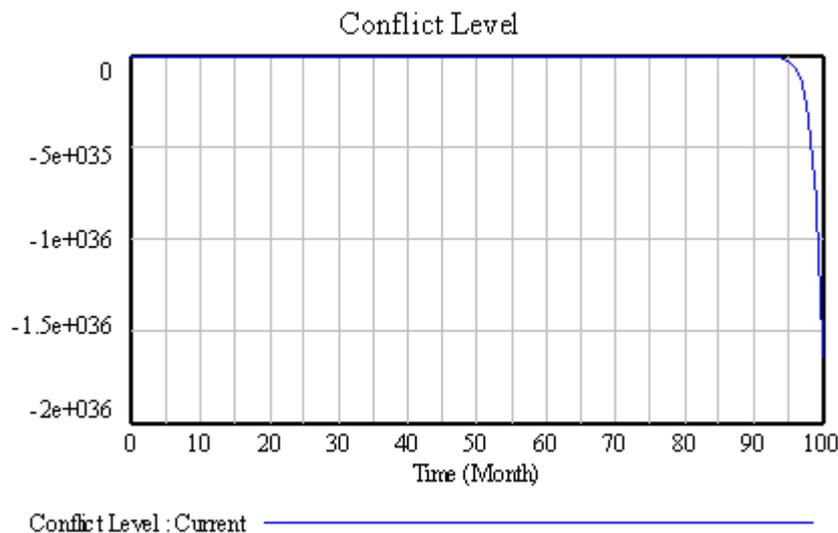
When choosing Community Size=20, the values are, of course, exactly the same as those generated in the previous equation. When entering the size of her community (n=500), however, conflict quickly spirals completely out of control (so much so, that the simulation program gave an overflow error when calculating the values of Conflict Level)

The next step is to model possible interventions which could reduce conflict so that it becomes manageable. Upon further reading the literature, Jane discovers an interesting paper which may offer an explanation for the differences in behavior in her case. It has been shown that when communities grow larger, they require well-designed segmentation strategies to deal with information overload, for example by splitting large fora into sub-fora (Ginsburg, 2001). As this notion of information overload is in agreement with what the community moderators have told her, Jane decides to adapt her model: instead of community size directly affecting conflict generation, *information overload* is the intermediate variable (Fig.12). For now, she decides to adopt a simple linear relation: a certain increase in Segmentation Level leads to a similar reduction of Information Overload. This approach makes sense, at least to start with, since splitting a forum in two (assuming each forum is led by a different moderator) roughly halves the number of messages which must be handled.



**Figure 12** The same model extended with an Information Overload dynamic model component

When the Segmentation Level equals 1, there is no segmentation, and Conflict Level spins out of control in exactly the same way as in the previous model. However, with a Segmentation Level set to 2, the Conflict Level gets drastically reduced. With Conflict generated =  $(10 + (\text{Conflict Level}/10) - (\text{Productivity}/9)) * (\text{Information Overload}/20)$ ; Community Size = 500; Information Overload = Community Size / Segmentation Level; Segmentation Level = 2 (Fig.13):



**Figure 13** Results of the simulation of the model with increased Segmentation Level

So, with only the smallest possible segmentation, conflict that has spun completely out of control becomes

completely manageable, at least given the assumptions in the models selected.

Jane now is confident that size matters in understanding conflict in large-scale online gaming communities, and decides to focus her initial research questions on testing the simulated relationships in great detail in a careful empirical study.

## References

- Andrews, D. (2002). Audience-Specific Online Community Design. *Communications of the ACM*, 45(4), 64-68.
- Axelrod, R. (1984). *The Evolution of Cooperation*. New York, Basic Books.
- 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*.
- Fernández, W. D., & Lehmann, H. (2005). Achieving Rigour and Relevance in Information Systems Studies: Using Grounded Theory to Investigate Organizational Cases. *The Grounded Theory Review*, 5(1), 79-107.
- Ginsburg, M. (2001). Growing Out of Its Skin: Principles of the Evolution and Extension of the Internet Chess Club, 1995. *Proc. of the 7th Americas Conference on IS (AMCIS 2001)*, pp. 1541-1521.
- 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.
- McNurlin, B. C. and R. Sprague Jr. (1989). *Information Systems Management in Practice*. London, Prentice-Hall.
- 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.
- Urquhart, C. (2001). An Encounter with Grounded Theory: Tackling the Practical and Philosophical Issues. In E. Trauth (Ed.), *Qualitative Research in Information Systems: Issues and Trends*, Idea Group, p114-140.
- 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.