A LEAGUE OF ITS OWN:
TOWARDS A NEW ONTOLOGY FOR SOCIAL SOFTWARE

Wim Bouman[1] and Tim Hoogenboom
Department of Business Studies
University of Amsterdam Business School
Roetersstraat 11, 1018 WB, Amsterdam, The Netherlands

Keywords
IS Theory and Design, Social Software, Ontological Foundation of IS

Abstract

When is it relevant, useful or appropriate to denominate a certain IT-based system as social software? IT-based systems can not meaningfully be qualified as social software by its design per se, nor by its functions. It is the actual manifestation in practice only that ultimately defines the qualification of a particular IT-based system. We define social software as IT-based systems, engaged by their users as an unfolding object for constructing and reproducing their social relations. Reapplying traditional IS logic to the design, development and deployment of social software might turn out to be devastating. IT-based systems become a thing to live with, and designers are liable for finding solutions that improve the way people interact with these IT-based artifacts.

[1] Corresponding author: Drs. W. Bouman, Department of Business Studies,
University of Amsterdam Business School, Roetersstraat 11, 1018 WB, Amsterdam, The Netherlands,
Phone: +31 (0)20 525 4266, Fax: +31 (0)20-525 417, E-mail: w.bouman@maatschap.org

1 SOCIAL SOFTWARE: A LEAGUE OF ITS OWN

Social software is a league of its own, and deserves to be treated in its own way in order to reap the full benefits for learning, knowledge management and connecting people within and between formal and informal social groups and organizations. This article deals with two main issues. First we deal with the issue of ontology: *When is it relevant, useful or appropriate to denominate a certain IT-based system as social software?* Our research indicates that an IT-based system can not meaningfully be baptized as social software by its design per se, nor by its functions. It is the actual manifestation in practice only – its social function of establishing joint enterprise, shared repertoire and mutual engagement – that ultimately defines the qualification of a particular IT-based system. Our ontological definition of social software, taking into perspective all our observations is: An IT-based system, engaged by its users as an unfolding object for constructing and reproducing their social relations. To elucidate this formal, explicit specification of the conceptualization of social software we provide a framework of archetypical IT-based systems further delineating the various basic schemes of information systems, knowledge management systems and social software systems. Secondly, we investigate the design consequences of the above premise: *What are the consequences, if any, of applying IS logic to the design, development and deployment of social software systems?* Applying the all too familiar logic from transactional information systems, meant to enable informing in a practice, to knowledge management systems, suited in learning context, has in our opinion been disastrous. Reapplying that same IS logic to the design, development and deployment of social software systems might turn out to be even more devastating. Two vignettes, one on the co-existence of formal and informal tools during a course at university and one on the subject of messaging in an large corporation, serve as an empirical illustration.

2 AN ONTOLOGY OF SOCIAL SOFTWARE SYSTEMS

We advocate a new ontological and epistemological approach towards social software systems. Ontology is the study of the nature of being, existence or reality in general, as well as of the basic categories of being and their relations. It is the formal, explicit specification of a shared conceptualization. We aim for a definition of social software that describes its essence of being as an IT-based system in the formative context of its surrounding practice. Furthermore, we aim to properly position social software systems relative to the basic categories of IT-based systems (e.g. information systems and knowledge management systems).

2.1 DEFINING SOCIAL SOFTWARE: THE STATE OF AFFAIRS

Social software is in need of its own theoretical and epistemic foundation, yet scientific literature is remarkably scarce for such a widespread real-life phenomenon. The current state of research and theory focuses primarily on the application of social software tools (most of which are the well-known publicly available social software services) in e-learning environments. In this perspective, social software is regarded a valuable contribution to social learning. Other scholars mainly proclaim the importance of this new development, provide often empirical, casuistic descriptions of social software applications while clarifying taxonomical issues. Social software is seen here primarily as a technology phenomenon, and research is aimed at engineering and improving social software solutions. Research into the fundamentals and foundations of social software seems almost non-existent.

The lack of fundamental research publications seems strange considered from an IS/IT perspective. In the era of the networking knowledge worker (Nardi, 2002; Wittel, 2001; Knorr Cetina, 1997), we need to understand and overcome the mismatch between the inherently social character of the knowledge worker's practices and the functionality-oriented tools that support these practices. Today's users of the often free public social software services are tomorrow's designers, managers, decision makers and (non-) users of business support systems.

In general, we notice a move towards all-embracing theoretical concepts in which auction management systems like eBay or syndication services like Listserv neatly fit into the social software categorization. It even seems that if a software application has any capability of communication or interaction, it is likely to be labeled social software. Mejias (2005), for instance, considers multiplayer gaming environments, discourse facilitation systems (a wide array from mail, chat to messaging and so on), moderated commenting systems (such as Slashdot), content management systems (blogs, wikis, document management), web annotation utilities, product development systems, peer-to-peer file sharing systems, selling or purchasing management systems, learning management systems, relationship management systems (like Friendster and Orkut), syndication systems (list-servs, RSS aggregators) and distributed classification systems (e.g. Flickr, del.icio.us) as social software. Dron (2006) also provides a categorical definition of social software, spanning blogs, wikis, link sharing systems, collaborative filters and other tools employing tagging, social recommendation and social navigation.

We also note a predominating IS/IT orientation towards social software. All common definitions inhibit metaphors of functionality, e.g. 'support', 'enabling', or 'use', thus centering the discourse around technology instead of social phenomena such as learning, knowledge or sociality.

Lawley's definition illustrates this IS/IT orientation: Social software is the term he uses to designate '*the use of computing tools to support, extend, or derive added value from social activity—including (but not limited to) weblogs, instant messaging, music and photo sharing, mailing lists and message boards, and online social networking tools'* (Avram, 2006:1).

Definitions of social software from a proto-theoretical viewpoint also seem to take a primarily technology angle. Social software has been '*loosely defined as software which supports, extends, or derives added value from, human social behavior – message-boards, musical taste-sharing, photo-sharing, instant messaging, mailing lists, social networking*' (Coates, as cited in Farkas 2005:1). Clay Shirky, an other prominent writer in the field of social software, according to Boyd (2006:15) *'(re)claimed the term 'social software' to encompass "all uses of software that supported interacting groups, even if the interaction was offline, e.g. Meetup, nTag, et cetera"*'. Shirky's claim was intentional, since he felt older terms such as 'groupware' were either polluted or a bad fit to address certain new technologies. Tepper (2003:19) states that '*Many forms of social software are already old news for experienced technology users; bulletin boards, instant messaging, online role-playing games, and even the collaborative editing tools built into most word processing software all qualify*'.

Research is often technology centered, whereas modern thinking would lead us to adopt a more post-human or post-social perspective. As shown by Pawson and Tilley (1997), it is not a physical artifact that is causal to people changing or adjusting their behavior in practice. Yet, the role that objects play in social associations has recently been recognized as increasingly important (Latour, 2005). Objects need to be 'emotional and intellectual companions' that help us organize our lives, sustain relationships and provoke new ideas (Turkle, 2007).

An intriguing question in this debate is whether an IT-based system itself acts as an object with agency, or that the IT-based system merely provides the social objects around which people weave their associations and movements. We assume the IT-based system to act as 'a thing to live with', a social object that unfolds in time by the interplay in a complex social system of designers, users, non-users and other actors, each having their own worldview. Stressing the role of practice and agency in an ontology for social software requires a valid theory into the resources and reasoning of people acting in their social contexts, a field of study in which philosophy, sociology and other social sciences excel above traditional IT related academic disciplines.

2.2 CRITIQUE AND CONCERNS

We have various concerns in the contemporary approach towards social software. First and foremost, we find the common definitions of social software too broad and of very limited use for theory and practice. Mejias (2005:2) acknowledges that '*as with all labels, there is some ambiguity and controversy over what kinds of things are supposed to be included under the 'social software' label, or how it differs from previous labels such as 'collaborative software', 'groupware', et cetera*'. Whereas we think that the above conceptualizations do not properly relate to the real world of events and practices (as we would certainly not consider eBay itself a typical social software system, for instance), they underline the importance of accepting that it is the actual manifestation in practice that determines whether a particular IT system is to be considered social software or not. After all, we have no stopping rule or scientific law that prohibits classifying eBay as a social software system. Yet, we agree with Dron (2006:904) who states that '*Some extend the definition to include virtually any software that supports social interaction, including newsgroups and email, but this stretches it beyond the point of usefulness'*. Indeed, if (nearly) everything is social software, the concept in itself looses practicality and scientific relevance.

Secondly, the focus is primarily on information related problems, such as information retrieval, metadata or other forms of information annotation, not – as we would suggest – on the social relationships affording new forms of togetherness. Although the informational aspects of social relationships are relevant for the debate at hand, we feel that this focus on IT and information gives only a very narrow view to the social practices in everyday social life. Social phenomena are neither fully described nor understood in terms of technology or information. The social reality of living together is both constructed and embedded (Pawson and Tilley, 1997), so we criticize any approach that neglects these rich social processes. Instead, we favor any approach that treats social phenomena from the viewpoint of actors in a network assembling their social relations or associations (Latour, 2005).

Thirdly, given its predominant preference for objectivistic, deterministic or mechanistic reasoning, IS/IT literature seems often to neglect the indeterminateness, or wicked nature, of social software design and development. As a practice and a science, information management struggles to understand the inherent social character of human relationships and tend to adopt either an undersocialized or an oversocialized conception (Granovetter and Swedberg, 2001). Adopting a simple, objectivist approach leads to the pitfall of thinking in terms of problems and solutions (Huizing, 2007a), instead of a infinite complex human, purposeful interaction where social behavior, social relations, social groups and social objects are constructed in a way that can not fully be understood, tested or stopped (Rittel and Webber, 1973; Buchanan, 1992).

Another symptom of deterministic reasoning is that often non-users are not accounted for. The focus in research is often on groups within the software system, not towards groups or social structures in real life. Perhaps a final consequence of such a mechanistic, objectivist *Weltanschauung* (Checkland and Poulter, 2006) is a tendency of objectifying human beings who in practice act purposeful, voluntary and knowledgeable as moldable ‘users’ – and act upon this concept of user instead of understanding and transforming the real world of events.

3 TOWARDS NEW UNDERSTANDINGS

The research program of which this article is a spin-off takes into account the need for a better ontological foundation of social software and counterbalances our critique and concerns.

3.1 RESEARCH APPROACH

This article stems from the Amsterdam Business School’s ongoing research enquiry into the relations between sociality as a human tendency, (the agency of) social objects, the constitution of social relationships and associations and the role of social software as a social object acting in or mediating the construction of social relations. We are intrigued by people choosing social network mechanisms, social objects and social software to create and maintain new forms of togetherness and sociality. Our area of research is the field of formal, complex modern organizations. To provide some background for our ontological approach, we briefly outline the key aspects from our research.

We depart from the proposition that not all relations qualify as social. Following Fiske (1992), we acknowledge that people are social by nature and that sociality is the human tendency to form or engage in social groups. That does not make all relations social; only relations in which people are prepared to adjust their behavior and attitudes to conform with the expectations of others are social in nature. To quote a well-known example; one might argue that a hired killer and his victim have a relation, yet one would not call this a social relation at all. Our study is in the infinite complex world of object-centered social relations.

The research approach is based on soft systems thinking and soft systems methodology. It can be considered learning for action (Checkland and Poulter, 2006), where researchers enter a social problem armed with a set of conceptual systems in a practice based ‘process of enquiry’. Our goal is not to offer solutions to people in a particular situation with problems, but to organize our learning and research process along the lines of developing rich pictures and relevant models of human, purposeful action and the mechanisms that can be triggered to bring change in this real life social world.

Our research approach is practice based; our analyses and systems models are at the level of practice (Giddens, 1984) and avoid the extremes of structure or agency (Cavanagh and Huizing, 2009).

As the main concepts of our study unfolded, the structure of lacks in the contemporary conceptualizations became apparent and the need for a better understanding of the concept of social software emerged. In creating rich pictures we learned that social software in itself is best understood as an epistemic device; as a conceptual model describing what we see in the real world not as a hard system that really exists. IT-based systems do exist, but social software systems do not exist as such; they appear and unfold themselves in use in their formative context only. An IT-based system (e.g. a wiki) that serves for a certain social group as a collaborative editing tool, might be the nexus of social activity or object-centered sociality for another group. The code, structure and functionality of the software may even be exactly the same.

3.2 A NEW ONTOLOGY FOR SOCIAL SOFTWARE

Following Checkland and Poulter (2006), we acknowledge that software systems are ‘systems serving systems’ that are used by sociable, reflexive and purposeful operating actors in a complex social context. IT-based systems live a life of their own; they exist only with the formative context of purposeful activities in the real world. At the same time, the system served is transformed by the new ideas and possibilities embedded in or provided by the software system. Following that logic, we do not wish to contribute to the ontological and epistemological development in IS by merely applying new labels for the sake of our specific argument.

Terms like information systems, social software and knowledge management systems are widespread in everyday life. Yet, in scientific research it seems not too self-evident to distinguish between these various appearances of IT-based systems[2]. Mostly, the broad term IS is considered to cover all applications of IT in organizational information systems. We hold that this is no longer appropriate as the social world moves away from the industrial era. It makes sense to draw sharp distinctions between the typical manifestations of information systems, knowledge management systems and social software systems. We need these archetypical schemes to better understand and deploy IT-based systems in our practices.

[2] For the purpose of this article we limit our archetypical models to administrative applications of IT. Other archetypical models, such as industrial, technical or embedded IT based systems, are intentionally out of scope, as these systems do not primarily serve social systems.

The distinction between the three archetypical models does not originate from the underlying IT functionality. One might legitimately argue that on a functional level a modern e-learning management system does not differ in essence from a social software application such as a wiki. Yet, from the perspective of its users the three manifestations of IT-based systems differ dramatically in their social application. People expect entirely different things from the IT-based systems they have at their disposal, and associate with these objects in completely different modes. Following Fiske (1992), one might hypothesize that the association with a traditional information system is based on a market pricing or authority ranking mode of sociality, whereas a social software system is engaged by its users in a mode of communal sharing or equality matching. And even more important; various IT-based systems serve in as well provide various formative contexts for human activity systems (Checkland and Poulter, 2006) in which they play a role.

Since people may and will expect different behavior or even agency from their IT-based systems, it is necessary to understand the various different characteristics of the three archetypical models mentioned. *Information Systems* provide and process information in a predefined context, like an organization or business process. The final objective is to have the information at disposal in order to better decide what (not) to do. *Knowledge Management Systems* exist for the purpose of capturing someone’s knowledge in a specific moment in time and to make it available to someone else in another place, time and context. The final objective is to (partially) answer a question previously unanswered to a person, or to refer to a source, person or organization that could be of help in the process of answering the question under study. *Social Software Systems* principally serve the purpose of extending or enriching one’s social relations, associations or network and to tap into its potential as a source of connectedness around various objects or issues.

From the perspective of the actor(s) involved, the difference is enormous. In Information Systems, for instance, one is concerned with the quality of information in terms of its correctness and timeliness as the information transaction at hand is usually to follow organizational or regulatory guidelines. It would be annoying at least to discover that invoices do not consistently contain the right data although they stem from the organization’s financial or ERP system. In Knowledge Management Systems, people are usually aware of the information asymmetry effects and thus take into account that no information in itself can – without sense making (Choo, 1998) and negotiation of meaning (Wenger, 1998) – ever provide a viable answer for the specific situation under study (Huizing and Bouman, 2002).

In Social Software Systems, information for the most part plays a role as a social object. Information serves as an ‘unfolding object’ (Knorr Cetina, 2001) for people to construct and maintain their social relations. People do not tend to take any information provided as a cold substance. In social software systems like Flickr, for instance, photos are engaged more as a trigger for social conversations, for creating and negotiating meaning and as a means for constituting and maintaining new forms of togetherness than as purely information stored for future reference.

3.3 SOCIAL SOFTWARE AS AN UNFOLDING OJECT

Our ontological definition of social software, taking into perspective all our observations is: *An IT-based system, engaged by its users as an unfolding object for constructing and reproducing their social relations*.

This conceptualization stresses a couple of points mentioned above. Firstly, it is the system’s formative context – the practice in which the system is assimilated – that defines its social nature, not the functionality. The fact that IT-based systems with similar functionalities are engaged differently is elucidated in Exhibit I. Secondly, a system is conceptualized as ‘a system serving systems’, an IT-based system that serves the needs and wants of other human purposeful activity systems (Checkland and Poulter, 2006). Thirdly, the conceptualization stresses a sociality-centered approach, in which actors are seen as purposeful, knowledgeable and voluntary participating (Pawson and Tilley, 1997). And fourth and foremost, it underlines an object-centered perspective, in which social software systems are treated as objects that co-constitute social relations and unfold their structure of lacks (Knorr Cetina, 2001) for practices with a structure of wants.

Exhibit I – Peaceful co-existence: Manifestations of technologies in the classroom

This exhibit describes the case of a university providing a formal e-learning system to support students, while at the same time students deploy a free blog service. It provides an excellent example of students accepting a formal tool in their formal course setting, while at the same time adopting a similar tool for their extensive social practices. Not even did these toolsets peacefully co-exist, they turned out to be complementary.

An educational course in economics provides Blackboard as an e-learning system, offering functionality from basic document storage and retrieval to interactive communication facilities such as discussion forums. The course has been given for almost ten years now, and a common observation is that students do not like to use these social functionality at all. Instead, they prefer their own means of support – often free tools. Next to the Blackboard e-learning environment, in recent years a blog service was chosen as a meeting place by students. This, at first, seems strange. From an IT point of view Blackboard and a Wordpress blog offer similar functionality, like posting and facilitating discussions in response to these posts. Traditional information systems thinking therefore might regard this redundancy as unwanted. In a way, it is; information about the course is stored in different places and one is never sure or can control that every participant is at the same level of information. As our experience with the course grew, we learned to see things in a different, social light.

Students mentioned that they saw their blog as a kind of ‘private study’, providing them a greenfield setting to freely experiment. Students blogged intensively and voluntary, to sharpen their minds among their peers about the theories and implications of the lectures. Their vivid discussions illustrate an unfolding practice of students entering a discourse to gain a more complete understanding of the theories. Although the blog was inspired by the lectures, it had no formal link with the university or classroom practices. The Blackboard environment, carrying the formal seals of the university, was shed by students. They conceived it as a formal extension of the class room, which left them afraid because thoughtlessness or deviant thinking would be easily notified by professors. Students wanted to avoid any backfiring of their comments, during lectures or exams.

It appeared that the use of the blog actually differs from the formal Blackboard system. Students actually posted their understandings on the blogs, while sharing their questions or asking fellow-students if they shared their views. The discussion section often served the unfolding of new (partial) representations of the blog post, while in between the original poster and fellow students thanked each other for contributing and spurred new food for thought. Furthermore, the blog also served the purpose of strengthening the subgroup processes by serving as a meeting place. Next to the knowledge sharing practices, the blog also served as means of communication. Students, for example, notified each other about interesting authors and arranged offline meetings to carry their discussions or assignments forward.

These activities contrasted sharply with the activities observed on Blackboard. The unfolding strengths of the blog remained largely absent within Blackboard. Students posted items of a rather definitive and complete nature. Looking back on their activity on Blackboard students explained that they wanted to maintain in the certain safe-zone, focusing on content value and applicability, to avoid attracting attention by towering above their peers. Blackboard was certainly not the place for the traditional CD – a yearly ritual compiling everyone's favorite songs – to be discussed.

The students' blog was actually thought of as something different, which led to other expectations regarding its use and interpretation. Students thought of it as a personal refuge to exploit personal opinions and discuss thoughts with their peers. They invited not only their classmates. Other kindred spirits, like students from other disciplines, bloggers or other visitors, were also welcome because of the peripheral boundaries of the blog. Students felt that the blog was actually a means to reinforce their social rankings within the class. Some students mentioned that they noticed others gaining an 'expert' or 'authority' status, due to their discussions or contributions. While the blog crossed or even ignored the boundaries of the course – making it also viable after ending the course – the Blackboard space was perceived as a tool suitable within the context of that course. Its predefined list of members, course students and teachers, reinforced its relatedness with the course. That students actually thought of Blackboard as a controlled medium was shown by the students' contributions. These appeared to be dictated by social desirability or being visible for the teaching staff, rather than a sincere effort to contribute to their own learning or the development of the group. Last but not least, students are still puzzled by the design theory that underlies the absence of a 'mail to all' function in the formally endorsed Blackboard implementation.

The lessons learned in this case indicate that, although functionality might be similar and from a traditional IS perspective having multiple ambiguous IT-based artifacts is considered squandering resources and efficiency, from a sociality-based perspective we need to accept the wickedness of social reality. Similar tools are interpreted and lived with – even by the same people in the same timeframe – as completely different social objects that can not and will not collide.

3.4 ARCHETYPICAL MANIFESTATIONS OF IT-BASED SYSTEMS

IT-based systems might appear the same in terms of their technical and functional capabilities, yet they are essentially unique for every situation in human life. We do not focus on the inherent properties but rather on how an IT-based system manifests itself in the social fabric of a practice. To further clarify our conceptualization of social software, we compare social software systems with two other archetypical manifestations of IT-based systems in the following table.

Archetypical manifestation	**Information Systems**	**Knowledge Management Systems**	**Social Software Systems**
System Rationale	Informing	Learning	Belonging
Ontological Stance	Hard systems thinking	Dependant on knowledge metaphor *(knowledge as an object versus knowledge as negotiated meaning)*	Soft system thinking
Epistemological Stance	Tame problems	Predictable problems	Wicked problems
Design logic	Designing by drawing	Designing as a process	Designing without a product
Form	Functionality driven	Pre authorized paths of reasoning	Dynamic, ‘unfolding’ objects
Function	Improving information supply and exchange	Matching questions and answers (via expert logic)	Triggering sociality and social relations in practice
Patterns of Reasoning	Technological deterministic	Asynchronous presentation and representation of knowledge disembedded in time and space	Sociality driven around social objects
Actor Model	Economic, Bounded rational actor	Transactional, knowledgeable actor	Social, Network embedded actor
Implementation strategy	Blueprinted implementation	Free choice adoption	Socially triggered voluntary participation
Concept of Usability	Being able to operate and to interact	Content value and applicability	Desirable in social context

Exhibit II – Archetypical manifestations of IT-based systems

The *Ontological Stance* of the three categories is dominantly dependant on the underlying concept of the system. In the Information Systems archetype, a hard systems line of thinking seems to underlie the general system characteristics. That means that in general, the system is made up of components that are considered as existing in real life, independent of a viewer's perspective. For instance, an invoice is an invoice in real life as well and has an empirical reference within the real world. In the Social Software Systems archetype, on the other hand, systems are considered soft systems, existing only in the actor's head for the purpose of organizing and categorizing their social life. Concepts like diggs, kudos, (on-line) community, clans, friends and groups have the empirical meaning that is shared amongst the participants only in their real-life social context. The ontological stance of the archetypical scheme of Knowledge Management Systems is located in between, depending on the assumed nature of knowledge. If knowledge is seen as an object (Huizing, 2007a), it tends to a hard systems stance (sometimes referred to as the Acquisition Metaphor as opposed to the Participation Metaphor (Sfard, 1998)). If, on the other hand, knowledge is seen as negotiated meaning stemming from learning processes (Huizing, 2007b), the ontological stance tends to be more soft systems thinking oriented.

The *Epistemological Stance* deals with how knowledge and related concepts are understood and scoped. In the Information Systems archetype, the knowledge issue is considered a tame problem of rather predicatively acquiring more and better information. Usually, the aim is for the right information for the right person on the right moment. Boisot (1998) refers to this model of learning as 'generative learning'. The underlying assumption is that there is a certain way of deciding what is right for whom and when. This assumption is rejected by wicked problem thinkers, who argue among others that wicked problems have no way of assessing 'true' or 'false' and have no stopping rules (Buchanan, 1992; Rittel and Webber, 1973). The Social Software Systems archetype takes a more subjectivist approach towards reality, as opposed to the more objectivist approach of the Information Systems manifestation. The difference is the proposition whether it could be possible to pinpoint a 'true' description of reality or that an objective reality does not exist but is constructed by every individual every moment in a social context. Knowledge Management Systems archetypes are suitable working models for circumstances in which informational or learning problems are either predictable in content or in process, e.g. in a doctor's diagnosis tool or an e-learning system. In that case, it is accepted that knowledge is always 'knowledge to someone' but that the processes leading to learning can be facilitated, moulded and triggered. Epistemologically, we hold that Social Software Systems are embedded and should adhere to the resources and reasoning of individuals in their social context, even more than Information or Knowledge Management Systems.

The *Design Logic* underlying the various manifestations is very different in nature. Design refers to the human activity of creating artifacts, to create change in manmade things (Buchanan, 1992; Jones, 1992). A design logic suitable for the Information Systems archetype is 'designing by drawing', a form of designing in which the designer can trust upon his blueprints to be a valid descriptor of a viable system. In fact, a certain amount of drawing before creating is efficient and necessary to establish the right design choices (which are more or less independent of interpretation in time). The design logic for Knowledge Management Systems manifestations is oriented to the process or the method. Here, not the desired outcome of the system is designed but the process of getting there. In general we observe that in knowledge management problem situations, the choice of tools becomes arbitrary compared to the processes of sense making and meaning negotiation ('this is a good way of sharing and preserving knowledge in our situation'). Designers of knowledge management systems are usually aware of the learning curves of people using the systems, and are able to create and adopt the design concepts over time. The Social Software Systems archetype needs a different design approach called 'designing without a product', meaning that designers depart first and foremost from a reasoning on intentions and (intended and unintended) consequences of a designed system. It has proven impossible to predict the social processes that lead certain groups to accepting a social software system as their tool of choice (Bouman et al., 2007). A viable design choice is to participate in a collective process of discovering the way IT-based systems are interpreted, used and abandoned. Whereas non-participation in Information Systems is almost a non-issue as the systems are endorsed by the organization if not mandatory (which, we do like to add, does not mean that it is always a pleasure to use them), non-participation in social context may be crucial in understanding and improve the systems' serving capabilities.

The *Form* that IT-based systems take, differs between the three manifestations. Information Systems archetypes usually have the shape of functionally oriented containers of functionality, e.g. for storage, handling, presentation or distribution of information. Form, for Information Systems, equals serving purpose. A search path in a taxonomy, or the predefined workflow are examples. Knowledge Management Systems provide answers or answer directions to questions (whether they are cognitive, conditional, or procedural) usually in a pattern with a more or less predefined logic reasoning. Form is relevant for purpose, since the order in which questions are asked or answers are given determines the rigor and relevance of the system. Social Systems archetypes have a non-linear form in which various functionality is available for the use by the systems participants. The non-linearity is a necessity for social systems, as participants themselves accept and adopt the (social) practices in which, how and when the designed functionality is used and unfolded.

The *Function* of an IT-based systems describes the particular set of functions or capabilities associated with its use. In order to serve the practice of informing, Information Systems archetypes diminish information asymmetries by relying on syntactically matching information demand and information supply in a mainly rigid fashion. Key is to provide an exhausting pool of information while leaving interpretation of the relevancy to the inquirer (Huizing and Bouman, 2002). In this archetype, humans reside outside the communications within the system, and solely act on the output produced by the system. To serve the (epistemic) practice of belonging, Social Software Systems archetypes enable for object-centered sociality, implying the signification of relations between objects and subjects that permanently unfold into another representation of that relation. Key is to provide relations that denote a kind of meaningful affiliation (Knorr Cetina, 2001). In the social software archetype, humans and technology mutually construe their output. The practice of learning is supported by Knowledge Management Systems manifestations, and predominantly focus on matching and answering questions. It transcends the functionality of pure exchange of information, by reckoning in context. This context can be based on case based reasoning from an objectivist perspective, or be based on a interplay of interpreting and answering questions from more subjectivist perspective.

The *Patterns of Reasoning* that underpin the archetypical models are particularly about the level of inherent dynamics. In the Information Systems archetype, a determinate snapshot of reality is contained in a definitive problem definition phase, yielding a precise solution (Buchanan, 1992). In this archetypical manifestation, technological determinism is the dominant pattern of reasoning. Inherent habitual routines are contained within IT-based systems and these automated routines impose the use or presentation of information to the users. Evolving contextual developments are continually staged into systems additions that are released. This release mechanism creates the impression that designers and users exercise a certain amount of control over their systems, yet one might also argue that during its lifetime the IT-based system predominantly shapes the resources and reasoning its environment according to the inherent constraints in the system and its underlying logic. In case of archetypical Social Software Systems manifestations, objects and humans mutually co-construct each other on the basis of their relation. This relation exceeds routine reasoning, as stated above, however, it actually constitutes an epistemic practice that continuously unfolds new representations of the design, use and operation logic of the software. By realizing these design and operation logics, new uses and forms of togetherness are emerging continuously, based on the relational interaction between object and subject.

The patterns of reasoning in the Knowledge Management Systems archetype are again somewhat located in between these two extremes of habitual routines and epistemic practices. Knowledge is seem to live in the heads of human actors, and the IT-based system is there to contribute in the asynchronous presentation and representation of knowledge disembedded in time and space, in such a way that knowledge of a certain person becomes in one form or other accessible to another person.

The three distinguished archetypical manifestations are characterized by *Actor Models* that differ considerably in their assumptions of the agency, knowledgeability and sociality of the actors involved. The Information Systems archetypes are grounded in rational or economic actor models. For satisfactorily exchange of information to take place any asymmetries are expected be nullified, which requires complete rationality by both the demanding and supplying actor (Huizing, 2007a). The Knowledge Management Systems archetype is founded on knowledgeable actor models, which accept that humans are bounded rational, are opportunistic experience communication limits (Huizing and Bouman, 2002). In the Knowledge Management Systems manifestation, actors are furthermore considered learning individuals in social groups (e.g. organizations) that invoke collective learning themselves. Learning inhibits an interplay between interpreting questions and translating answers. IT-based systems are considered inapt to capture the complete richness of contextual differences, which requires knowledgeable actors to make up for residuals. Sociable actor models underlie the Social Software Systems manifestation models. The practice of belonging relies on a actionable, sociable or networked actor model, for as IT-based systems are not inherently social, actors have an emergent want to associate to one another or to objects. IT-based systems can only trigger mechanisms of slumbering networking deficiencies, by the grace of networked actor models. Voluntary participation, based on economic and social motivations and constraints, is a basic premise in this archetype.

An *Implementation Strategy* fitting the Information Systems archetype is to synchronize the logic set out in the design phase with the logic for the implementation phase. The determinedness of variables initially captured in the design should ideally be prolonged into a blueprinted and controlled implementation, directing its participation and use. People will accept a new IT-based system if it proves efficient and useful. In case of the Knowledge Management Systems manifestation, the use of content is constrained to situational dependencies and thus its implementation or rather adoption is one of free choice and voluntary participation. People are expected to contribute to and use the IT-based system if a clear added value to the professional practice is shown. Moreover, knowledge itself is regarded an organizational asset that can and should be exploited. Effective implementation strategies take care of not mixing up the organizational interests in knowledge acquisition and participation with the human mechanisms of protecting and mystify personal capabilities, competences and knowledge.

In case of Social Software Systems archetypes effective implementation will follow the logic of voluntary participation (Lakhani and Von Hippel, 2003) and the unfolding nature of objects (Knorr Cetina, 2001). People engage in social relations and associate with groups or objects according to the norms, behavior and constraints of their formative context. Voluntary participation can not be designed, it can only be designed for. A way of designing for fluid social patterns and agency, is to exploit the unfolding nature of social objects. Effective implementation accepts that a social object is never finite; objects have a structure of lacks that does or does not fit in with the structure of wants of the actors involved – which leads participants to engage in a open conversation, practice, discourse around that imperfection. In order to be 'a thing to live with' Social Software Systems therefore have to provide thought provoking objects over time and allow participants to shape their own social needs within the collective constitution of the social associations.

The *Usability* aspects differ between the archetypical manifestations of Information Systems, Knowledge Management Systems and Social Software Systems. In the archetype of Information Systems, usability seems to thrive on practical acceptability. An IT-based system should for instance provide ease of operation and relevant interaction in case of errors or additional information (Nielsen, 1993). Usability is a concept that is seen as referring to the system and its capabilities. In Knowledge Management Systems, usability is seen as the quality of information objects. Within Knowledge Management Systems manifestations the value of content and its applicability within a professional practice are seen as central concepts. A knowledge management system is regarded usable by its users if the system's contents are appropriate and applicable. In case of archetypical Social Software Systems, usability depends on the degree to which other actors in the social context choose to use or neglect an IT-based system and whether use or non-use has social implications, e.g. contributes to one's identity. The concept of usability is stretched beyond functionality towards sociality; it is not the inherent properties of technology nor the inherent properties of the content contained and presented but socially constructed acceptance that determines whether a system is perceived usable.

Exhibit II's treatise further illustrates the ontological delicacies surrounding archetypical manifestations of IT-based systems (Information Systems, Knowledge Management Systems and Social Software Systems) and logically substantiates our argument that applying an information systems perspective towards design and use of social software could prove disastrous.

4 DESIGN CONSEQUENCES: NO RIGHT TO BE WRONG

In contemporary scientific literature little is known about the consequences, if any, of applying IS logic to the design, development and deployment of social software systems. Yet, designers simply can not dodge the effort to uncover the theoretical and epistemic foundations of social software. Otherwise, they might fall into the traps of the ill-conceived and possible hazardous reapplying of mechanistic IS/IT patterns of thinking towards a new, potentially vital and exciting class of business applications.

The development of knowledge management systems has already been fallen prey to a often dogmatic reapplication of IS reasoning. The focus lies increasingly on communicative functionalities, in the form of the capturing, storing, and transferring of objects, through which the fixed meanings of objects could be transferred from an active sender to a passive receiver (Huizing, 2007). Information systems rely on being able to objectify the real world, like invoices in an ERP system, and wrapping it in metadata to become more manageable. In order for information systems to function properly this objectification of real world should be confined to circumstances having definitive conditions.

Knowledge management systems too are often based on reapplying this same old logic. In theorizing on the knowing organization (Senge, 1990; Nonaka and Takeuchi, 1995), supportive IT-based solutions are devised on the same IS/IT patterns of thinking. This kind of thinking promotes objectification – e.g. capturing, explication, storing – of knowledge as being necessary to make the fabric visible and thus manageable within organizations (Alavi and Leidner, 1999). Nonetheless, this premise rests on a flawed view on social exchange, which assumes that objects of knowledge contain fixed meanings that can be replicated during transfer between sender and receiver, without being subjected to noise or interpretation.

In social practices, according to wicked problems thinking (Buchanan, 1992), designers have no right to be wrong. Therefore, we argue that in designing an IT-based system that should manifest as a social software system, designers need to take into account the practices in which the designed object is to live, the role it plays in the social fabric and its capability to enter and alter social relations by changing the resources people have at their disposal and the reasoning with which they make sense of their surroundings (Pawson and Tilley, 1997; Choo, 1998).

The inherent wicked nature of social software and its design tend to reject blueprinted approaches and 'designing by drawing' logic (Jones, 1992). Linear approaches of uncovering the definitive conditions that need will need to be addressed to solve a determinate problem will not match up to the indeterminacy of social practices.

The social is inherently complex, and no singular path to salvation exists. Instead, design and its products for the social are just as ephemeral as the social locus in which they are situated. We can only have partial understanding of social problems, and by engaging in these problems, new understandings will unfold that will require adjustments in earlier designs and products (see Exhibit III for an illustration).

The prevailing patterns of IS/IT reasoning seem to permeate present-day design logic in social software. Boulos and Wheeler (2007:147) maintain a functionalist perspective of the rich nature of social software, claiming that these technologies represent a '*revolutionary way of managing and repurposing/remixing online information and knowledge repositories [...] in comparison with the traditional Web1.0 model*'. Avram (2006:6) holds similar convictions, stating that '*Weblogs, wikis and social network services have as their core purpose knowledge sharing, and RSS feeds have made sharing even easier'* and purports that social software could be a intuitive case wrap for disclosing information in an innovate way. Both examples illustrate a functionalist account in the design and application of social software, neglecting the software's abilities to trigger sociality, by means of forming and servicing social relations and associations.

Exhibit III – Messaging: My colleague is not my buddy

This exhibit describes the case of a company aiming for the implementation of an enterprise version of a popular instant messaging client. Instant messaging is a form of real-time communication between two or more people based on typed text.

From an IT point of view, it appeared a logical next step in office automation to add instant messaging to the functionality available at the desktops and laptops of the employees. Also, it seemed self-evident to leverage the existing investments in IT to create a corporate messaging environment. Following a traditional IS logic, the approach took the following direction. At first the contacts and groups were identified and prefilled on the basis of the organization's design. Thereafter rules were established to regulate naming conventions, customizations and plug-ins. Compliance with security policies meant no external communication. And a standard messaging client would be provided, excluding multiple messenger clients. As a result, the implementation did not gain momentum.

From our analysis, we concluded that people have radically different expectations of the behaviors of these IT-based systems. A messenger client is something we have at home, and are free to use in the way we like. We choose our own avatars, our own names en our own e-mail addresses, and with the greatest of ease rename and recategorize our buddies, or ban or ignore them. People have no difficulty in using corporate systems in the prescribed way. For social software services that can also be used freely in public and private spheres this appears to be a different story. Company policies restraining these freedom of use are even considered infringements of individual degrees of freedom.

A different approach would have been helpful, balancing the organization's needs and the employees' needs and wants and – last but not least – practices and expectations. This approach should reckon the wickedness of social reality, e.g. people chose their buddies freely, yet their colleagues are imposed. This seems to fit the essence of human social relations; a colleague is a more formal relation, based on market pricing or authority ranking (Fiske, 1992), which differs from a buddy, which relies on an communal sharing model.

Also, the corporate structure does not translate easily to and individual's segmentation of social networks and peer groups. The conventions in formal organizational (information or learning) systems are predefined and pre-authorized. From an organizational perspective this rigid embeddedness makes sense, because it does not seem to efficient or pleasant to continually renegotiate meaning of common business terms over and over. This sharply contrasts with the common freedom, voluntary participation and identity (or imago) control found on a global scale.

Social networks and groups almost per definition extend organizational borders and are by means of peripheral membership, brokering and so on interconnected in a loose but defined way. Whereas it made sense in this case to restrict the flow of communication and information with the outer world as the company has to regulate communications with many customers in the business area, this negatively influences the usability of the IT-based system intern. People see no harm to ask a friend for a solution by messenger, in fact this is what they are used to do. Evidently, from a formal viewpoint on organizing, this is undesirable.

Like other IS implementations, companies adopts a policy of standardization which is only too sensible from a cost perspective. It does not make sense to maintain various ERP systems, nor does it to sustain more than one CRM system; not only cost wise, but also from an informational standpoint. The choice for the closed implementation was therefore downright, yet this is at odds with the way people are able to connect their buddy lists from various sources in integrated clients.

All in all, the false start of the project meant rethinking the concept of the instant messenger software in that specific context. The lesson learned is that as a formal business organization one can not groundlessly adopt an IT-based system already hugely successful in the everyday social life of people and expect it to seamlessly integrate in the formal, bounded business context. One needs to rethink the concepts of identity, communication, document exchange and so on; but first and foremost one needs to have an in-depth understanding of the place the company might be allowed to gain within the existing web of social relations of the employees and their contacts and contexts.

Designers who fail in realizing their responsibility to create and maintain proper design theories on what will work for whom run the risk of squandering an organization’s resources. According to Rittel and Webber (1973), in a social context designers do not have the luxury to exhaustively test their design with hard data. They are expected to get things right the first time. The continuation of the social practices and relations may stall or get hurt if a designer holds an oversimplified design theory, or no theory at all. Designers are not necessarily liable for finding a true solution, but rather finding a solution that improves the conditions for the people interacting with these IT-based artifacts.

5 CONCLUSION

The promise of social software – reaching beyond information transactions per se towards the constitution of social relations and associations in complex social groups – is widely recognized and considered a potential value added for future business applications. Intra-organizational adoption of these services seems only a matter of time.

IT-based systems can not meaningfully be baptized as social software by its design per se, nor by its functions. Instead, it is the actual interplay of sociable, knowledgeable and participating actors and social objects in the practice only that ultimately defines the qualification of a particular IT-based system. Our ontological definition of social software is: *An IT-based system, engaged by its users as an unfolding object for constructing and reproducing their social relations*. IT-based systems are to become ‘things to live with’.

The ontological approach presented in this article attempts to open up our perspective on three manifestations of technology: Information Systems, Knowledge Management Systems and Social Software Systems. Its contribution to the field of study is in the potential avoidance of making the same mistake twice. In the application of sociality-driven applications of IT, we consider it crucial to its success to treat the implementation of social software according to its own sociality-driven logic. Neglecting the basic premises of sociality means not only squandering scarce resources. Designers have no right to be wrong and unrightfully interfere with social relationships and associations. Social software is truly in a league of its own, and calls for both a better understanding and a better application.

ACKNOWLEDGEMENTS

The authors thank the members of De Maatschap at the Amsterdam Business School for their valuable contributions to this article, especially René Jansen, Ard Huizing and Mark Nijssen.

BIOGRAPHIES

Wim Bouman and Tim Hoogenboom are Research Fellows at the Amsterdam Business School at the University of Amsterdam in The Netherlands. Both are engaged in the Amsterdam Business School's ongoing research enquiry into sociality, social objects and social software. They are also the authors of the ICIS 2007 Best Paper Award for their article *The Realm Of Sociality: Notes on The Design of Social Software*. We encourage and appreciate feedback via w.bouman@maatschap.org and t.hoogenboom@maatschap.org.

REFERENCES

Avram, G. (2006), 'At the Crossroads of Knowledge Management and Social Software', *Electronic Journal of Knowledge Management*, vol. 4, no. 1, pp. 1-10.

Boisot, M. (1998), Knowledge Assets: Securing Competitive Advantage in the Information Economy, Oxford University Press, New York, USA.

Bouman, W., Hoogenboom, T., Jansen, R., Schoondorp, M., Bruin, B. de, Huizing, A. (2007), 'The Realm of Sociality: Notes on the Design of Social Software', *Conference Proceedings 28th Annual International Conference on Information Systems* (ICIS), Montreal, Canada.

Boyd, D.M. (2006), 'The Significance of Social Software' in Burg, T.N. and Schmidt, J. (eds.), *BlogTalks Reloaded: Social Software Research & Cases*, Norderstedt, pp. 15-30.

Buchanan, R. (1992), 'Wicked Problems in Design Thinking', *Design Issues*, Vol. 8, No. 2, pp. 5-21.

Cavanagh, M. F. and Huizing, A. (2009), 'From 'what we know' to 'what we don't know' – A Practice-based Approach to Organizational Learning', *Conference Proceedings Fourth International Conference on Organizational Learning, Knowledge and Capabilities* (OLKC), Amstelveen, The Netherlands.

Checkland, P.B., and Poulter, J. (2006), Learning for Action - A Short Definitive Account of Soft Systems Methodology and its use for Practitioners, Teachers, and Students, John Wiley and Sons Ltd., Chichester, United Kingdom.

Choo, C.W. (1998), The Knowing Organization: How Organizations Use Information to Construct Meaning, Create Knowledge, and Make Decisions, Oxford University Press, New York, USA.

Dron, J. (2006), 'Social software and the emergence of control', *Conference Proceedings Sixth IEEE International Conference on Advanced Learning Technologies* (ICALT), Washington, DC, USA. pp. 904-908.

Farkas, M.G. (2007), Social Software in Libraries: Building Collaboration, Communication, and Community Online, Information Today, New Jersey, USA.

Fiske, A.P. (1992), 'The Four Elementary Forms of Sociality: Framework for a Unified Theory of Social Relations', *Psychological Review*, vol. 99, no. 4, pp. 689-723.

Giddens, A., The Constitution of Society, Polity Press, Cambridge, UK, 1984

Granovetter, M. and Swedberg, R. (2001), *The Sociology of Economic Life* (second edition), Westview Press, Colorado, USA.

Hoogenboom, T., Kloos, M., Bouman, W. and Jansen, R. (2007), 'Sociality and learning in social software', *International Journal of Knowledge and Learning*, vol. 3, no. 4-5, pp. 501-514.

Huizing, A. and Bouman, W. (2002), 'Knowledge and Learning, Markets and Organizations: Managing the Information Transaction Space', in Choo, C.W. and Bontis, N. (eds.), *The Strategic Management of Intellectual Capital and Organizational Knowledge*, Oxford University Press, New York, USA.

Huizing, A. (2007a), 'Objectivist by Default: Why Information Management needs a new Foundation', in Huizing, A. and De Vies, E.J. (eds.) *Information Management – Setting the Scene*, Elsevier Press, Oxford, United Kingdom, pp.xx-xx.

Huizing, A. (2007b), 'The Value of a Rose: Rising above Objectivism and Subjectivism', in Huizing, A. and De Vies, E.J. (eds.) *Information Management – Setting the Scene*, Elsevier Press, Oxford, United Kingdom, pp.xx-xx.

Jones, J.C. (1992), *Design Methods – Second edition*, Wiley, New York, USA.

Knorr Cetina, K. (1997), ‘Sociality with Objects: Social Relations in postsocial Knowledge societies’, *Theory, Culture & Society*, vol. 14, no. 4, pp. 1-30.

Knorr Cetina, K. (2001), ‘Objectual Practice’, in Schatzki, T., Knorr Cetina, K. and Savigny, E. von (eds.), *The Practice Turn in Contemporary Theory*, Routledge, London, pp. 175-188.

Lakhani, K.R., and Von Hippel, E. (2003), How Open Source Software works: ‘Free’ User-to-User Assistance, *Research Policy*, vol. 32, no. 6, pp. 923-943.

Latour, B. (2007), Reassembling he Social: An Introduction to Actor-Network-Theory, Oxford University Press, New York, USA.

Mejias, U. (2005), ‘A nomad’s guide to learning and social software’, *The Knowledge Tree*, vol. 7.

Mejias, U. (2006), ‘Teaching social software with social software’, *Innovate*, vol. 2, no. 5.

Nardi, B., Whittaker, S. and Schwarz, H. (2002), ‘NetWORKers and their Activity in Intensional Networks’, *Journal of Computer-supported Cooperative Work*, vol. 11, no. 1-2, pp. 205–242.

Nielsen J. (1993), *Usability engineering*, Academic Press, Boston, USA.

Pawson, R. and Tilley, N. (1997), *Realistic Evaluation*, Sage Publications, London, United Kingdom.

Rittel, H., and Webber, W. (1973), ‘Dilemmas in a General Theory of Planning’, *Policy Sciences*, vol. 4, pp. 155-169.

Sfard, A. (1998), ‘On two Metaphors for Learning and the Dangers of Choosing just One’, *Educational Researcher, 27*, 4–13.

Swedberg, R. and Granovetter, M. (2001), Introduction to the Second Edition, in Granovetter, M. and Swedberg, R. (eds.), *The Sociology of Economic Life (second edition)*, Boulder / Oxford: Westview Press.

Tepper, M. (2003), ‘The Rise of Social Software’, *netWorker*, vol. 7. no. 3, pp. 18-23.

Turkle, S. (2007), *Evocative Objects: Things We Think With*, MIT Press, Cambridge, USA.

Wenger, E. (1998), *Communities of practice: Learning, meaning and identity*, Cambridge University Press, Cambridge, USA.

Wittel, A. (2001), ‘Toward a Network Sociality’, *Theory Culture & Society*, vol. 18, no. 6, pp. 51-76.