Designing the Undesignable: Social Software and Control

Jon Dron
University of Brighton, UK // Jon.dron@brighton.ac.uk

ABSTRACT

Social software, such as blogs, wikis, tagging systems and collaborative filters, treats the group as a first-class object within the system. Drawing from theories of transactional distance and control, this paper proposes a model of e-learning that extends traditional concepts of learner-teacher-content interactions to include these emergent properties of the group. It suggests that this feature of social software can facilitate an approach to e-learning that is qualitatively different from and capable of significantly augmenting traditional methods, with especial benefits for lifelong learners and those outside institutional boundaries. The paper goes on to explore some of the dangers and issues that need to be addressed in order for this new model to fulfill its promise, and to suggest a framework of principles to be used by designers of educationally-oriented social software.

Keywords

Social software, Transactional distance, Control, Web 2.0, Self-organization

Introduction

This is a paper about the potential of social software in education. Social software treats the group as a first class object within the system (Allen, 2004). It is becoming embedded in all aspects of our online lives, from the recommendations of Amazon to the photo sharing of Flickr to the ordering of search results in Google. Social software such as blogs, wikis, link sharing systems, collaborative filters and other tools employing tagging, social recommendation and social navigation are increasingly a part of the fabric of an e-learning environment. In this paper I argue that they have characteristics that strongly distinguish them from earlier software for e-learning, providing particularly notable benefits to informal and lifelong learners. I describe a theoretical foundation which explains the potential benefits of social software in the context of transactional distance and control, extending earlier frameworks for e-learning to account for the emergent control of the group. Drawing on this foundation, I offer a set of ten principles to be employed by designers of social software environments for e-learning.

Transactional control

Michael Moore's theory of transactional distance suggests that the relative amount of dialogue and structure is what determines 'distance' in learning, rather than physical separation between learner and teacher (Moore, 1986). It is implied by the theory and independently confirmed by Saba and Shearer, that the two are inversely interdependent (Saba & Shearer, 1994). The greater the structure, the lesser the dialogue, and vice versa. To this mix, Moore adds a third dimension of autonomy. Fully autonomous learners need neither dialogue nor structure to learn. Like dialogue and structure, autonomy is seldom absolute. Others have refined this concept, noting that there is a big difference between autonomy as a psychological trait, and the varying degrees of autonomy of the learner to choose a path within a given learning context (Brockett & Hiemstra, 1991; Candy, 1991).

Moore's theory has been widely used and verified in a number of contexts (Chen, 2000; Dron, Seidel, & Litten, 2004; Lowe, 2000; Saba & Shearer, 1994; Stein, Wanstreet, Calvin, Overtoom, & Wheaton, 2005), but suffers from a fuzziness and inconsistency that troubles some (Garrison, 2000; Gorsky & Caspi, 2005).

I have proposed a different but intimately connected model, a subset of transactional distance that I have called *transactional contro*l (Dron, 2006). Transactional control is concerned with choices. In traditional education, whether online or offline, some choices are made by teachers and some by learners. The extent to which those choices are dictated by particular individuals determines their level of transactional control at any moment. Transactional control theory does not aim to replace transactional distance theory as it says nothing significant of the psychological gap between learner and teacher, but it helps to explain some of its dynamics. Structure equates to teacher control, dialogue to negotiated control, and autonomy to learner control. At any point in a learning trajectory, transactional control will vary. It is concerned with the choices that determine a change in trajectory, not those that

follow from the intrinsic logic of the transaction nor from the extrinsic constraints which mould it. However, we shall see that there are other ways that control can arise and other entities that can wield it.

Figure 1 illustrates how transactional control captures the dynamics of transactional distance, but not the trait of autonomy nor the closeness felt by the learner to the teacher.

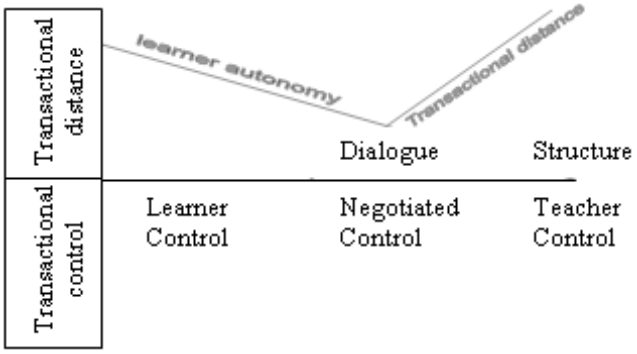

Figure 1. transactional control mapped to transactional distance

Transactional control is always on a continuum between two extremes, changing with each choice we or someone/something else makes. No matter how much a teacher may control the transaction, the learner can always choose not to pay attention, to disagree (if only internally) or perhaps to interrupt. Similarly, the learner must almost always delegate control at some level. Sometimes this may be a result of extrinsic or intrinsic constraints. More often than not, the simple fact of being a learner implies a lack of knowledge and consequently requires control to be delegated to one who possesses that knowledge and is willing to communicate it, whether directly or mediated through a book, web page or computer program.

Control occurs at a range of scales. As in nature and the development of cities, the large influence the small more than vice versa (Brand, 1997). The smaller the scale, the more constrained the choices become: each level of constraint applies further control that adds to the previous. When we reach the point of making decisions about which word to use next, or whether to read the next word in the sentence, it is barely a choice at all. Choices *are* made at this level, but the constraints overwhelm the choices most of the time. The fact that a choice is small does not necessarily reflect its consequences. If I make the small choice to continue walking when that leads me into the path of a speeding juggernaut, the effects may be significant.

A change in learning trajectory occurs when it represents a departure from the path dictated by intrinsic and extrinsic constraints. The smaller the choices become, the more there is an element of subjectivity in deciding what constitutes a change in trajectory. What may be a natural continuation for some may seem like a change of tack to another.

Most learning transactions tend towards control by either the learner or, more often, the teacher. From a learner perspective, being given control without the power to utilize it effectively is bad: learners are by definition not sufficiently knowledgeable to be able to make effective decisions about at least some aspects of their learning trajectory. On the other hand, too much teacher control will lead to poorly tailored learning experiences and the learner may experience boredom, demotivation or confusion. Dialogue is usually the best solution to the problem, enabling a constant negotiation of control so that a learner's needs are satisfied (Garrison & Baynton, 1987). Unfortunately, dialogue is expensive and becomes more so the fewer participants are involved. The ideal would be to allow the learner to choose whether and when to delegate control at any point in a learning transaction.

Social software

Social software is a relatively new and poorly defined concept. Systems such as del.icio.us, Flickr, YouTube, MySpace and Ning are unequivocally accepted as instances of the genre, and most writers are happy to accept that

wikis and blogs (or at least, collections of blogs) are likewise forms of social software. So too are technological approaches such as collaborative filtering/ recommender systems, shared tagging and social navigation. Some extend the definition to include virtually any software that supports social interaction, including newsgroups, instant messaging and email (Mejias, 2005), but this stretches it beyond the point of usefulness. One useful way to distinguish social software from earlier forms of mediated communication is that, compared with, say, chat rooms, discussion forums and mailing lists, it scales very well, gaining strength from large numbers: while earlier forms of computer mediated communication tend to become overwhelmed once a certain optimum number of participants has been exceeded, social software (with some provisos) just gets better and better.

Social software is organic and self-organizing, underpinned by dynamics that parallel natural processes. It is evolutionary, replicating the successful and diminishing or killing the unsuccessful (Shirky, 1996). It is stigmergic: signs left in the environment communicate with others who leave further signs in the environment (Bonabeau, Dorigo, & Theraulaz, 1999). It has an emergent structure, formed from bottom-up control rather than top-down design.

Social software has recently become popular in online learning, but its roots in this context can be traced back to the early nineties (e.g. (Ackerman, 1994; Boder, 1992; Davis & Huttenlocher, 1995)). Collaborative filters have long been used for learning (e.g. (Dron, Mitchell, Siviter, & Boyne, 1999; Recker, Walker, & Wiley, 2000; Terveen, Hill, Amento, McDonald, & Creter, 1997)) and social navigation techniques have been applied in various educational contexts (e.g. (Brusilovsky, Chavan, & Farzan, 2004; Dron, Mitchell, Boyne, & Siviter, 2000; Kurhila, Miettinen, Nokelainen, & Tirri, 2002; Miettinen, Kurhila, Nokelainen, & Tirri, 2005; Vassileva, 2004)). Blogs and wikis, as well as other forms of social software, are becoming commonplace (e.g. (Barker, 2005; Downes, 2004; Dron, 2003; Mejias, 2005)). The tools seem, intuitively, to offer value to learners, allowing the social construction of meaning and relatively effortless collaboration in new and interesting ways. Social software presents the learner with options, implicitly or explicitly recommending links and paths as a result of the behavior of an indefinite number of others. This can range in scale from the tight, linear control of the text of a wiki to the social navigation signposts of different font sizes for tags in del.icio.us, to the vague and amorphous choices implicit in links and trackbacks between blogs or Google's PageRank algorithm (Brin & Page, 2000). It therefore may be seen as fulfilling a teacher role of providing control over the learning trajectory. This control is not the result of a single individual's determination of a trajectory, but is an emergent property of the interactions of the group as a whole.

Modes of interaction in e-learning

Anderson describes six modes of interaction in online learning – teacher-student, teacher-content, teacher-teacher, student-student, student-content and content-content (Anderson, 2003). In social software, the group is a first class object that has an existence in its own right, mediated through the environment just as interactions between individuals are mediated. A further four interactions are therefore significant: student-group, teacher-group, content-group and group-group.

Student-group: In a social software environment, the learner is a part of the group mind, influencing yet influenced by it. This dual role makes the notion of control very fluid.

Teacher-group: The role of the teacher in social software systems is potentially less significant than in conventional e-learning environments, despite the fact that use of such software may be initiated at a larger scale by the teacher and he or she may determine goals and outcomes, shaping the ecology in which it resides. For example, when I presented links to my own lecture notes along with other resources through a social link-sharing system they were less popular than some of the others (Dron, Mitchell, Boyne, & Siviter, 2000). Lately I have taken to providing my notes in the form of a wiki, to which students contribute and correct, sometimes because I ask them explicitly, sometimes because they spot mistakes, wish to add more depth or seek clarification. I have used blogs for several years, gaining many benefits, notably including the fact that students teach each other more than I teach them (Dron, 2003).

While the teacher may have a relatively small role in the detailed form that a social software system may take, social software can tell the teacher a great deal about the group: their preferences, their interests, their needs, their weaknesses, their strengths.

Content-group: Social software typically leads to an emergent structure that is not planned by an individual, but which emerges from the individual actions of members of the group. In turn, this affects the individual actions of its members. Thus, the content is a reification of the group's behavior, and the group's behavior is at least partially a consequence of the content.

Group-group: Open standards such as RSS and web services make the exchange of information between social software systems a simple and ubiquitous feature. It is common to link together such diverse services as Google Maps, Flickr images, blogs, del.icio.us tags and so on to create new 'mashups.' Group to group interactions make blogs into social software: an individual blog is little more than a web page with annotations/discussions attached. However, when combined through blogrolls, trackbacks, common tags and embedded feeds, the space becomes the blogosphere, rich and interconnected, with clusters, social groupings and two-way links combining to create an emergent and intricately structured community.

Social software and transactional control

Because of its emphasis on the group, social software occupies an interesting space in terms of transactional control and, consequently, transactional distance, providing both user-control and the delegation of control simultaneously. Recursively, structure arises as a result of dialogue (perhaps implicit) which in turn influences dialogue. Negotiation of control leads to a reified structure, to which control may be delegated. A dependent learner can choose to be controlled, while a more autonomous learner can take more control. Both may influence and be influenced by the structure that ensues, to whatever extent that they find comfortable.

Benefits to lifelong learners

Social software can allow learners to choose whether to control their learning or to delegate that control to the group. In principle, then, it appears to offer the best of both worlds, assisting dependent learners through the provision of structure, yet enabling autonomy at any point. It is significant that this structure does not necessarily arise through the intercession of teachers or instructional designers. In a world where continual learning is the usual state of being and a necessary condition of successful existence, yet where the cost of traditional institutional learning is high, both in money and time, this is a strong benefit. Importantly, the structure that emerges through group interaction constantly evolves and changes, adapting to the changing needs and interests of the group it serves. This brings further advantages over traditional institutional forms of learning and teaching. For a range of reasons, some related to personalities and interests, some to the inevitably viscous institutional processes, it is the norm for the subject matter of institutional learning to slip a little behind the needs of industry, especially at undergraduate level and in fast-changing practical subjects such as computing, engineering and medicine. This is not to deny the importance of research-led teaching which moves ahead of industry on many occasions, but even then, the practical benefits of such subjects are typically less immediate for most learners. Driven by the needs of the group, it is easier for content to adapt quickly to more relevant subjects.

Like many others, if I wish to know what is the latest and most important research going on in my area, I am more inclined to turn to blogs and wikis than to journals and conferences. Traditional fora lag months or even years behind the current state of the art, whereas social software may tell me what happened yesterday or even today, and to participate in its development. Scientific breakthroughs are no longer the preserve of journals and conferences. Some social journals, such as PLoS One, or the older but still innovative JIME have exploited this process, blending the benefits of rigorous peer review with the strengths of a multitude of eyes and a plenitude of brains. Breaking news is no longer the domain of the established press as the likes of Digg become more relevant and responsive than traditional media could ever be. Google is often the first port of call for those seeking understanding of an unfamiliar topic and Wikipedia is often the second. Social software dominates our online landscape, so it is very important that we should explore ways to better exploit its benefits and avoid its pitfalls. Even small improvements in the effectiveness of such environments in meeting learning needs could bring immense value to lifelong learners for whom they are an increasingly significant source of information.

The darker side of social software

There are still some significant issues to be addressed if social technologies are to fully fulfill their exciting educational promise. Notably, the structure generated through social software may not be useful or pedagogically sound. Few systems employed with educational intent take this into consideration, allowing whatever structure that emerges to guide users. The CoFIND system is an exception, explicitly using two forms of tagging, encouraging not only the tagging of resources according to topic, but also with pedagogical metadata described as 'qualities,' which are then used to supply implicit and explicit ratings that feed back through stigmergic social navigation cues.

Many of the larger social software sites have millions of users, all of whom are contributing to the overall shape of the system. While the formation of small-world clusters makes these variegated spaces, they are often too large for significant parcellation to occur – it is one large evolutionary landscape, without the small peninsulas, continents and islands that deform the natural evolutionary landscape to allow diverse species to develop. It is significant that many (perhaps most) educational uses of such software make use of smaller systems which can adapt more easily to the needs of the group. Others (CoFIND again) intentionally parcellate the landscape so that smaller niches can develop.

Crowds are only wise when the individuals of which they are comprised are unaware of the choices others make (Surowiecki, 2004). With awareness of the behavior of others, the influence of a few (usually those who make the first contributions) is disproportionately large, shaping the behaviors of those who follow. Systems such as collaborative filters that amalgamate independently collected rankings are less susceptible to these effects than those employing social navigation or simple aggregation, but even very large collaborative filters such as Google are inherently stigmergic (Gregorio, 2003). They are susceptible to the Matthew Principle, whereby the rich get richer while the poor get poorer. The most common solution to this problem in institutional learning is to use such software as part of a learning ecology, often within specific taught courses, thereby situating them in a framework that is somewhat controlled by a tutor (e.g. (Brusilovsky, Chavan, & Farzan, 2004; Miettinen, Kurhila, Nokelainen, & Tirri, 2005)). However, if such structure is being used and generated by autonomous lifelong learners and a tutor is not in control, the positive feedback loop can run out of control.

Most social software is susceptible to intentional attack, whereby a malevolent or mischievous individual or group can bend the system to its purposes. For example, until recently, a search on Google for 'miserable failure' would point to George W. Bush's biography, as a direct result of the manipulation of backlinks by many people. Google have adjusted their algorithms to inhibit such attacks, but it is likely that, unless top-down control is exercised, flaws will again be discovered and exploited. Similar issues have been observed with social software used in education (Dron, Mitchell, Boyne, & Siviter, 2000; Vassileva, 2004). Intentional behavior can affect a virtual environment much as it can in nature and, much as in nature, the virtual environment can be polluted and corrupted. This remains an issue that is poorly resolved, with most existing solutions requiring some top-down editorial control.

Ten design principles for educational social software

There are many ways that social software can fail to address the needs of learners. If it is to be successful, there are ten principles that I have identified which make success more likely (Dron, 2007). The list is no doubt incomplete, but without adherence to each of these principles it is unlikely that social software will be successful in self-organizing for the benefit of learners. These principles are highly interconnected, each feeding from or informing one or more others and two in particular (connectivity and scale) are more meta-principles than guidelines.

The principle of adaptability

The principle of adaptability requires that we must try to build small services that interoperate, that we must build to connect using open standards and, where possible, we should build as open source, so that others may adapt and evolve systems to suit local needs.

Single-purpose software is like what Stuart Brand describes as 'magazine architecture' – beautiful perhaps, fit for purpose, but ultimately disposable as needs change and technology marches on (Brand, 1997, p. 52). Systems must be built based on the assumption that they are here today and gone tomorrow, especially those that deal with the constant flux of needs that emerge in lifelong learning. This implies an architecture of small, service-based,

interoperable systems, employing mashup technologies like RSS, web services, FOAF (friend of a friend), SSO (single sign-on), as well as stalwarts of the old-guard of designed learning technologies such as SCORM, IMS-LD and IEEE-LOM. A particularly promising technology is ELF (the E-Learning Framework) which specifies web services targeted at e-learning (JISC, 2004).

Open source approaches are ideally suited to adaptability as they allow changes to be made at a local level, which can fit the needs of particular groups.

Big, comprehensive commercial social software systems such as MySpace and FaceBook adhere to standards, at least to the extent that they can use and, in limited ways, share information with other sites. However, the kind of monolithic approach they embody is not to be recommended if we are to build truly adaptable systems. Such systems are the vacuum cleaners of the social world, sucking in other technologies but giving relatively little back.

The principle of evolvability

The principle of evolvability is that we must build deferred systems (Patel, 2003), systems whose structure is not fixed, systems that can change after the software designer has left the building.

Not only should the systems themselves be fluid and capable of change, the structures within the systems should be capable of change and evolution. Indeed, in a perfectly evolvable system, the rules of change themselves should be capable of evolving. This means creating systems that are not only adaptive but also adaptable. Evolution, by which I mean replication with variation combined with survival of the fittest, can be seen on most social sites. Tags in tag clouds may be seen as battling for prominence on an evolutionary landscape, with fitness decided (usually) by popularity. Wikis offer a rich evolutionary landscape where content is continually revised, deleted, refined and mutated to adapt to the needs and knowledge of the community (whose knowledge is itself revised and mutated as a result). Ideas in blogs are replicated and adapted as memes spread through trackbacks, annotated links, comments and blogrolls.

Systems that employ mashups may select between competing components, which consequently evolve at a higher level. A particularly successful and pure application of this principle is employed by Ning, where whole social applications are replicated and modified to suit changing needs and adapt to different ecological niches. Through a multi-dimensional matrix of tag clouds, these applications are red in tooth and claw as they fight for prominence in these constantly evolving ecological niches. Ning is a single site, but if it were more distributed and open, it would offer a superb model for social software design.

The principle of parcellation

The principle of parcellation requires that we must build systems in which there are distinct, ideally hierarchical ecological niches that are only weakly or occasionally connected with each other. Where possible, such niches should emerge rather than be imposed by a designer.

Ning's innovative use of tag clouds enables groups of applications to compete to some extent separately from other groups, which is very much in keeping with one of the prime drivers for evolution. Evolution works fastest on relatively small, isolated populations (it was, for example, the different species on the Galapagos Islands that inspired Darwin's theory), with changes propagated through the occasional rare event such as the formation of isthmuses through falling water levels, changing ice-cover or the chance mingling of ecosystems (Calvin, 1997; Jones, 1999). If systems become too large then innovations and changes tend to be swamped by the prevalent majority, and diversity is stifled (Gould, 1978).

The benefits of diversity and rapid adaptation brought about by parcellation are of special significance when social software is used for e-learning, improving its capacity to adapt quickly and to cater for diverse learning needs. Tags and topics/categories can help to provide some element of parcellation, but too often they simply chop up a large space into a number of equal small ones. In natural systems, there are richly connected hierarchies of parcellation: islands act as delimiters for mountains, that constrain the trees, that constrain the shrubs, that provide the boundaries

for the beetles, whose guts act as defining spaces for countless bacteria and other bugs. Intermingling occurs at several different scales, but the hierarchical boundaries offer great scope for local adaptation.

In a large social site such as Digg, the parcellation provided by tags is perhaps insufficient to allow the novel or minority interests to influence the majority, so a further top-down taxonomy has been introduced to help make sense of the mass of communication and information. To some extent this is a kludge. If we are to encourage evolution, we need to build systems that are far more variegated and diverse, where small ecological niches can form within the wider learning environment and where the landscape can be shaped, not just by system designers, but by the crowd. Without this, we introduce what Dennett (Dennett, 1995) describes as a 'skyhook' – an invented intervention from outside the system which, in a social software environment, seems a superfluous piece of top-down design.

Various approaches can assist the process of bottom-up emergence of parcellated structure. Elgg, for instance, allows creators of social spaces to emphasise specific key tags to create a kind of tag aristocracy, in which some tags are more important than others. All the various kinds of metadata in CoFIND are created by end users and their combinations form intentionally clustered micro-environments, albeit with a hierarchy that is currently limited to only two or three levels.

Whether imposed from the top down, as a kind of landscape gardening, or from the bottom up, as in a jungle, parcellation can be used to good effect if we wish to encourage the development of pedagogically useful structures. Top-down creation of structural parcellating features offers opportunities to teachers to regain some of the control that is inevitably lost when using social software, while bottom-up methods allow groups of learners to achieve more control themselves.

The principle of trust

The principle of trust makes it necessary for us to build the means to reliably identify reliability in people and resources, to protect ourselves from harm and to do this without resorting to top-down constraints.

Trust in an e-learning system is central to its success. There are many aspects of trust that are significant here, including: trust that it will work, trust that it is secure, trust that the information provided is reliable, trust that one will be supported, or at least not abused. As we move away from a central model of control, where policing comes from the top down, alternatives need to be sought so as to provide the learner with the necessary security to learn. Some, such as Piczo, take a simple if draconian approach by not giving a search facility, making it difficult or impossible to discover information about others except by explicit invitation. Although the site has many other innovative features to compensate, this is clumsy and counters some of the benefits of social software. Elgg takes a far better approach: its greatest strength as a tool for education lies in its elegant and hugely flexible authorization mechanisms. Access to every object in the system, be it a file, a blog posting or even a comment may be controlled in the minutest detail by its owner. This goes a long way towards dealing with issues of intentional attack and allows trust to be given at any level that a learner feels comfortable with. Authorisation arises from the bottom up and users can exclude those that they do not trust. The social networking features built in allow inhabitants to explore each other's profiles and social networks, to discover people with similar interests, to help to identify who is and is not trustworthy. Again, every bit of information can be revealed to as small or as large a population as the learner feels comfortable with. A less sophisticated but similarly intentioned mechanism is employed in my own Dwellings system. Once again, parcellation is important as trust is often more easily maintained in relatively small communities, where individuals are known.

Because we rely on the machine to mediate on behalf of the group, it is important that we trust its mechanisms and algorithms. If the results are helpful, then this will happen naturally, but even so it would sometimes be helpful to expose the machinery to the users, to give them control over the ways that information is aggregated and displayed, to create what Kay calls 'scrutable' systems (Kay, 2006).

The principle of stigmergy

The principle of stigmergy is that we should use signs to guide, but not to constrain, and to enable mechanisms to destroy those signs when they are no longer needed or are harmful.

People are influenced by signs left in the environment, whether direct (e.g. the presence of other people) or indirect (e.g. the size of specific tags in tag clouds). In nature, this kind of effect is known as stigmergy, a word originally coined to describe the nest building behaviour of termites, where pheromones in the saliva of termites collecting and dropping balls of mud act as signals to others to drop their balls of mud, collectively leading to complex cathedral-like structures through simple local communication processes. Much of the structure that is formed through dialogue-like processes in social software makes use of such social navigation cues, including things like the ordering of results in Google, the ubiquitous tag cloud in most social software and the density of messages in forums. Stigmergy is the primary means by which the group-mind is both revealed and formed. But stigmergy is a two-edged sword which, out of control, can fall prey to the Matthew Principle, whereby the rich get richer while the poor get poorer. This leads to runaway positive feedback loops like bank runs and devastating swarming behaviour. It is therefore vital to ensure that evolutionary or other feedback mechanisms (such as link fading, or other variations on pheromone evaporation) are in place that will prevent such out-of-control behaviour. Parcellation is again helpful here as it can limit the scope for damage caused by the Matthew Principle.

The principle of context

The principle of context is that, when building social software systems, we must consider the entire virtual ecosystem in which they reside and remember that they are only a part of a much greater whole.

If a system is not used with the intent to learn or assist learning, then the odds are that it will evolve into something different. It also exists in an environment which contains many other tools, social structures and hierarchies, and conflicting demands. If social software is not fit for its purpose, alternatives will be found – the continuing overwhelming popularity of email in the face of competing technologies is a testament to this (Anderson, 2006).

Context can also constrain. If the surrounding structures dictate an ethos of top-down control (notably in institutional learning) then the conflicting demands of teacher-imposed instruction and group-led direction may lead to a distorted, two-headed monster that gets consumed in its struggle to find a direction. A tutor who wishes to maintain control will be lost and confused in an environment which is controlled by the crowd.

The principle of constraint

The principle of constraint requires us to be aware of the constraints that we build into our systems and to use them to enhance learning, much as an architect influences use of a building through the placement of walls, windows and doors.

The degree to which system designers and administrators exert control over the system largely dictates whether its form will be determined by the group or by the designer. Sometimes this will be determined by context: blogs used in the service of traditional instruction will be very different from those that record a learning journey undertaken by an individual with personal motivations, for instance.

Decisions taken by system designers, on the other hand, profoundly affect the potential forms that may arise. At a coarse but far from trivial level of affordance, for example, if a system provides a facility for uploading images easily, the system will have more pictures one that does not. Affordances can also affect form and socialization. For instance, a system that allows personalization will lead to very different sense of ownership than one that does not. Interaction design plays a major role here: for instance, an always-on instant messenger will encourage different patterns of use compared with one that can only be reached through several intentional clicks through submenus. The inevitability of constraint should be capitalized upon, to encourage structures to evolve that are pedagogically useful. For example, CoFIND's use of qualities (pedagogic metadata) helps to bring focus on educational goals, while Elgg's fine-grained bottom-up user-controlled authorisation mechanism helps to build the trust needed in a learning community. Even when using a system that does not offer such features, it is possible for a designer to provide constraint by pre-seeding the system with content, tags and other metadata – this technique has been used by WikiProfessional, for example, which attempts to overcome concerns about reliability by starting with a large, reliable database culled from peer-reviewed sources. This uses the natural structural scaling where precedence, whether by design or as an accident of history, acts as a controlling influence on what follows.

The principle of sociability

The principle of sociability is that attention must be paid to the total system's capacity to enable social presence and communication and that, where possible, this should be embedded throughout.

Features of the software that enable communication and which encourage a sense of social presence are essential glue to hold together the structures that form, to negotiate subtleties of meaning, so socially construct knowledge. The group is an unintelligent beast despite being created out of intelligent agents, and the system must support direct interaction between people if they are to learn effectively. Comments on blogs and other shared resources, discussion forums, wiki discussion areas and so on can provide the fine-grained communication that individuals need to learn, diminishing the psychological gulf that transactional distance implies. Having said that, it is also important to remember that the environment is not simply a piece of software: for most of us, email is a click or even a mouse gesture away, and there are many alternative communication technologies available. This takes us back to the principle of adaptability: it is better to build small systems that slot together and interoperate than monoliths that attempt to be everything to every person.

The principle of connectivity

The principle of connectivity is that nothing should exist in isolation, everything should influence everything else, much as the beat of a butterfly's wing might affect the weather in another part of the world.

An over-riding principle that relates to most of the other principles is that of making connections: social software is entirely built upon connections between people, systems and subsystems. In particular, this means that an effective design for social software implies that a change anywhere in the system may have an impact, whether microscopically small, large, or anywhere in between, on anything and everything else in the system. Because structure is determined by behaviour, therefore all behaviour should have an effect. This is the primary reason that social software systems may only be influenced, not controlled. If there are parts of the system that do not have such connections, it may be a sign of top-down design intruding. This is not necessarily a problem as such (control may be what is sought), but should be viewed with suspicion by the designer if it arises unintentionally.

The principle of scale

The principle of scale is that we must be aware of the large and the small in our social systems, and ensure that, where possible, the large should arise out of the small in an endless iterative cycle of renewal.

The other central principle the underpins all the rest is the importance of scale. Whether thinking of hierarchical parcellation, degrees of constraint, the context in which the system resides or the overall design, the influence of the big and slow over the small and fast provides one of the key structural features that makes social software work, or fail. Scale effects often rely on an even higher-level structural feature of *priority*: one of the main reasons that the large influences the small is that it provides the context in which the small exists. By definition, therefore, the large exists prior to the small. On the other hand, while the small and fast seldom have much influence as individual system elements, the combined effects of many small elements can be vast: locust swarms and mob behaviour are obvious examples, but equally the spread of memes in a population or even simple stigmergic effects such as termite mounds or ant trails can bring about significant structural change to the environment. Social software capitalizes on this effect to bring about changes to large scale structures as a result of small-scale interactions, which in turn affect the small-scale interactions in a constant iterative cycle of growth and dynamic change.

Conclusions

Social software in e-learning offers great potential pedagogic and practical benefits, both through the amplification and creation of social ties, and through allowing learners to choose to choose whether they to control or be controlled in a learning transaction. This accounts to some extent for the enormous popularity of social systems like Google and Wikipedia which may fast be becoming the dominant learning tools of this first decade of the twenty-first century.

Perhaps this should be re-christened *g-learning* (for group-learning or Google-learning). It is, however, important that we understand the dynamics of social software and are aware of its potential weaknesses. A self-organizing environment is not necessarily an effective *learning* environment. Out of control, the wisdom of crowds can too easily become the stupidity of mobs, or be hijacked by those with other agendas. The ongoing challenge for developers and users of social software in e-learning is therefore to build systems and processes in which the structures that develop are capable of being pedagogically sound and supportive of learning communities. The ten principles described in this paper offer a starting point for discussion and experimentation, but they are undoubtedly a stepping stone towards a more refined, richer understanding of ways that environments may be built that encourage learning to occur, not through the intentional acts of individuals, but through the combined intelligence of the group.

A far fuller discussion of the issues and ideas discussed in this paper may be found in (Dron, 2007).

Websites and systems discussed in this paper

CoFIND: http://www.cofind.net
Del.icio.us: http://del.icio.us
Digg: http://www.digg.com
Dwellings: http://dwellings.cofind.net
Elgg: http://elgg.net
Facebook: http://www.facebook.com
Flickr: http://www.flickr.com
Google: http://www.google.com
JIME: http://www-jime.open.ac.uk/
MySpace: http://www.myspace.com
Ning: http://www.ning.com
Piczo: http://piczo.com
PLoS One: http://www.plosone.org/home.action
Slashdot: http://Slashdot.org
Wikipedia: http://www.wikipedia.org
WikiProfessional: http://www.wikiprofessional.info
YouTube: http://www.youtube.com

References

Ackerman, M. S. (1994). *Answer Garden: A Tool for Growing Organizational Memory.* Unpublished PhD, Massachussetts Institute of Technology.

Allen, C. (2004). Tracing the evolution of social software, Retrieved June 7, 2007, from, http://www.lifewithalacrity.com/2004/10/tracing_the_evo.html.

Anderson, T. (2003). Modes of interaction in distance education: Recent developments and research questions. In M. G. Moore & W. G. Anderson (Eds.), *Handbook of Distance Education* (pp. 129-146). New Jersey: Lawrence Erlbaum Associates.

Anderson, T. (2006). Social Software Applications in Formal Online Education. *Paper presented at the The 6th IEEE International Conference on Advanced Learning Technologies*, July 5-7, 2006, Kerkrade, The Netherlands.

Barker, P. (2005). *Potential Uses for Weblogs in Electronic Course Delivery.* Paper presented at the E-Learn 2005, October 24-28, 2005, Vancouver, Canada.

Boder, A. (1992). The process of knowledge reification in human-human interaction. *Journal of Computer Assisted Learning, 8*, 177-185.

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*, New York: Oxford University Press.

Brand, S. (1997). *How buildings learn*, London: Phoenix Illustrated.

Brin, S., & Page, L. (2000). The Anatomy of a Large-Scale Hypertextual Web Search Engine. Retrieved June 7, 2007, from, http://www-db.stanford.edu/pub/papers/google.pdf.

Brockett, R. G., & Hiemstra, R. (1991). A conceptual framework for understanding self-direction in adult learning. In R. G. Brockett & R. Hiemstra (Eds.), *Self-Direction in Adult Learning: Perspectives on Theory, Research, and Practice,* New York & London: Routledge, Retrieved June 7, 2007, from, http://home.twcny.rr.com/hiemstra/sdilch2.html.

Brusilovsky, P., Chavan, G., & Farzan, R. (2004). Social Adaptive Navigation Support for Open Corpus Electronic Textbooks. *Paper presented at the AH 2004*, August 23-26, Eindhoven, The Netherlands.

Calvin, W. H. (1997). The Six Essentials? Minimal Requirements for the Darwinian Bootstrapping of Quality. *Journal of Memetics, 1*, Retrieved June 7, 2007, from, http://williamcalvin.com/1990s/1997JMemetics.htm.

Candy, P. C. (1991). *Self-direction for Lifelong Learning: A comprehensive Guide to Theory and Practice*, San Francisco: Jossey Bass.

Chen, Y.-J. (2000). Transactional Distance in World Wide Web Learning Environments. *Innovations in Education and Teaching International, 38*(2), 327-338.

Davis, J. R., & Huttenlocher, D. P. (1995). Shared Annotation for Cooperative Learning. *Paper presented at the International Conference on Computer Support for Collaborative Learning*, October 17-20, 1995, Bloomington, IN, USA.

Dennett, D. (1995). *Darwin's Dangerous Idea*, Harmondsworth: Penguin.

Downes, S. (2004). Educational Blogging. *Educause Review, 39*(5), 14-26.

Dron, J. (2003). The Blog and the Borg: a Collective Approach to E-Learning. *Paper presented at the E-Learn 2003*, November 7-11, 2003, Phoenix, Arizona.

Dron, J. (2006). The Teacher, the Learner and the Collective Mind. *AI & Society, 21* (1-2), 200-216.

Dron, J. (2007). *Control and Constraint in E-Learning: Choosing When to Choose*, Hershey, PA: Idea Group.

Dron, J., Mitchell, R., Boyne, C., & Siviter, P. (2000). CoFIND: steps towards a self-organising learning environment. *Paper presented at the WebNet 2000*, October 30-November 4, 2000, San Antonio, Texas, USA.

Dron, J., Mitchell, R., Siviter, P., & Boyne, C. (1999). CoFIND- an experiment in n-dimensional collaborative filtering. *Paper presented at the WebNet 99*, October 24-30, 1999, Honolulu, Hawaii.

Dron, J., Seidel, C., & Litten, G. (2004). Transactional distance in a blended learning environment. *ALT-J, 12*(2), 163-174.

Garrison, D. R. (2000). Theoretical Challenges for Distance Education in the 21st Century: A Shift from Structural to Transactional Issues. *International Review of Research in Open and Distance Learning, 1*(1), Retrieved June 7, 2007, from, http://www.irrodl.org/index.php/irrodl/article/viewFile/2/22.

Garrison, D. R., & Baynton, M. (1987). Beyond Independence in Distance Education: The Concept of Control. *American Journal of Distance Education, 1*(3), 3-15.

Gorsky, P., & Caspi, A. (2005). A Critical Analysis of Transactional Distance Theory. *The Quarterly Review of Distance Education, 6*(1), 1-11.

Gould, S. J. (1978). *Ever Since Darwin- Reflections in Natural History*. Burnett.

Gregorio, J. (2003). Stigmergy and the World-Wide Web. Retrieved June 7, 2007, from, http://bitworking.org/news/Stigmergy/.

JISC. (2004). *The e-Learning Framework (ELF)– Summary*, Retrieved June 7, 2007, from, http://www.jisc.ac.uk/uploaded_documents/elf-summary7-04.doc.

Jones, S. (1999). *Almost Like a Whale*, London: Doubleday.

Kay, J. (2006). Scrutable Adaptation: Because We Can and Must. *Paper presented at the AH 2006*, June 21-23, 2006, Dublin, Ireland.

Kurhila, J., Miettinen, M., Nokelainen, P., & Tirri, H. (2002). Use of Social Navigation Features in Collaborative E-Learning. *Paper presented at the E-Learn 2002*, October 15-18, 2002, Montreal, Canada.

Lowe, W. (2000). Transactional distance theory as a foundation for developing innovative and reactive instruction. *Educational Technology & Society, 3*(1), 1-3.

Mejias, U. A. (2005). A Nomad's Guide to Learning and Social Software. *The Knowledge Tree*, Retrieved June 7, 2007, from, http://knowledgetree.flexiblelearning.net.au/edition07/html/la_mejias.html.

Miettinen, M., Kurhila, J., Nokelainen, P., & Tirri, H. (2005). OurWeb - Transparent groupware for online communities. *Paper presented at the Web Based Communities 2005*, February 23-25, 2005, Algarve, Portugal.

Moore, M. G. (1986). Self-Directed Learning and Distance Education. *Journal of Distance Education, 1*, Retrieved June 7, 2007, from http://cade.athabascau.ca/vol1.1/moore.html.

Patel, N. (2003). Deferred System's Design: Countering the Primacy of Reflective IS Development With Action-Based Information Systems. In N. Patel (Ed.), *Adaptive Evolutionary Information Systems* (pp. 1-29). London: Idea Group Publishing.

Recker, M. M., Walker, A., & Wiley, D. A. (2000). An interface for collaborative filtering of educational resources. *Paper presented at the 2000 International Conference on Artificial Intelligence*, June 26-29, 2000, Las Vegas, USA.

Saba, F., & Shearer, R. L. (1994). Verifying key theoretical concepts in a dynamic model of distance education. *The American Journal of Distance Education, 8*(1), 36-59.

Shirky, C. (1996). In Praise of Evolvable Systems. *ACM Net_Worker*, Retrieved June 7, 2007, from, http://www.shirky.com/writings/evolve.html.

Stein, D. S., Wanstreet, C. E., Calvin, J., Overtoom, C., & Wheaton, J. E. (2005). Bridging the Transactional Distance Gap in Online Learning Environments. *American Journal of Distance Education, 19*(2), 105-119.

Surowiecki, J. (2004). *The Wisdom of Crowds*, London: Little, Brown.

Terveen, L., Hill, W., Amento, B., McDonald, D., & Creter, J. (1997). PHOAKS: A System for Sharing Recommendations. *Communications of the ACM, 40*(3), 59-62.

Vassileva, J. (2004). Harnessing P2P Power in the Classroom. *Paper presented at the ITS 2004*, August 30-September 3, 2004, Maceio, Brazil.