PP2SS: From the Psychology of Programming to Social Software

Professor Marc Eisenstadt

Knowledge Media Institute, The Open University

Abstract. Social Software is software which amplifies or otherwise leverages human social behaviour. Common examples include message boards, reputation-based music file sharing, photo-sharing, instant messaging, blogs, wikis, and generic social networking hubs such as Friendster, LinkedIn, and Tribe.Net.
Interacting with other people not only forms the core of human social and psychological experience, but also lies at the centre of what makes the internet such a rich, powerful and exciting collection of knowledge media. We are especially interested in what happens when such interactions take place on a very large scale – not only because we work regularly with tens of thousands of distance learners at the Open University, but also because it is evident that being part of a crowd in real life possesses a certain 'buzz' of its own, and understanding the nature and power of group interactions poses a natural challenge. Different nuances emerge in different user contexts, so we choose to investigate the contexts of work, learning and play to better understand the trade-offs involved in designing effective large-scale social software for multiple purposes.
How does the Psychology of Programming fit into this? Right now, it doesn't: but it should. I believe that PPIG has much to offer the Social Software community, and vice versa. In this talk, I will outline four ways in which a symbiotic PP/SS relationship could grow:

1. SS -> PPIG: Existing social software tools can at the very least 'service' the PPIG community. Community members can provide mutually-referential blog entries describing their research, group wikis, and use RSS newsfeed aggregators and trigger alert services to act in a primary news dissemination capacity. Rapid interchanges can take place using a variety of synchronous social software tools. Right now, such tools are regarded as 'fringe', which is absurd: in the rest of the world, they are mainstream.
2. PP -> SS: Psychology of programming ideas can be of great benefit to the social software community. For example, current software environments underlying blogs and wikis are often extremely frustrating to use, typically because the deploy the wrong affordances or clash with the user's own mental models of the underlying virtual machines. This is 'home territory' to PPIG, and is therefore a ripe area for cross-fertilization.
3. PP(SS): Social software usage (as opposed to tool design) is a great domain of study for the PPIG community. For example, discourse patterns across 'duelling blog entries', and the ebb and flow of conversation in blogs, wikis, and instant messaging, is a ripe topic for investigation.
4. SS(PP): Applying a social software perspective to the very act of programming is another means of fostering synergy. Indeed, collaborative programming environments and paradigms, popular in the open source and MOO worlds, already familiar to many in the PPIG community, could be re-examined

in the light of the latest tools and interaction styles emerging in the social software world.

The talk will conclude with a view of what a PPIG workshop ought to look like in 2010.