A Few Thoughts on Engineering Social Machines

[Extended Abstract]

Markus Strohmaier
Knowledge Management Institute and Know-Center
Graz University of Technology
Graz, Austria
markus.strohmaier@tu-graz.ac.at

ABSTRACT

Social machines are integrated systems of people and computers. What distinguishes social machines from other types of software systems - such as software for cars or air planes - is the unprecedented involvement of data about user behavior, -goals and -motivations into the software system's structure. In social machines, the interaction between a user and the system is mediated by the aggregation of explicit or implicit data from other users. This is the case with systems where, for example, user data is used to suggest search terms (e.g. Google Autosuggest), to recommend products (e.g. Amazon recommendations), to aid navigation (e.g. tag-based navigation) or to filter content (e.g. Digg.com). This makes social machines a novel class of software systems (as opposed to for example safety-related software that is being used in cars) and unique in a sense that potentially essential system properties and functions - such as navigability - are dynamically influenced by aggregate user behavior. Such properties can not be satisfied through the implementation of requirements alone, what is needed is regulation, i.e. a dynamic integration of users' goals and behavior into the continuous process of engineering.

Functional and non-functional properties of software systems have been the subject of software engineering research for decades [1]. The notion of non-functional requirements (softgoals) captures a recognition by the software engineering community that software requirements can be subjective and interdependent, they can lack a clear-cut success criteria, exhibit different priorities and can require decomposition or operationalization. Resulting approaches to analyzing and designing software systems emphasize the role of users (or more general: agents) in this process (such as [1]). i* for example has been used to capture and represent user goals during system design and run time.

With the emergence of social machines, such as the WWW, and social-focussed applications running on top of the web, such as facebook.com, delicious.com and others, social machines and their emergent properties have become a crucial infrastructure for many aspects of our daily lives. To give an example: the navigability of the web depends on the behavior of web editors who are interlinking documents, or the usefulness of tags for classification depends on the tagging behavior of users [2]. The rise of social machines can be expected to fundamentally change the way in which such properties and functions of software systems are designed and maintained. Rather than planning for certain system properties (such as navigability, usefulness for certain tasks) and functions *at design time*, the task of engineers is to build a platform which allows to influence and regulate emergent user behavior in such a way that desired system attributes are achieved *at run time*. It is through the process of *social computation*, i.e. the combination of *social* behavior and algorithmic *computation*, that desired system properties and functions emerge.

For a science of social machines, specifically understanding the relationship between individual and social behavior on one hand, and desired system properties and functions on the other is crucial. In order to maintain control, research must focus on understanding a wide variety of social machine properties such as semantic, intentional and navigational properties across different systems and applications including - but not limited to - social media. Summarizing, the full implications of the genesis of social machines for related domains including software engineering, knowledge acquisition or peer production systems are far from being well understood, and warrant future work. For example, the interactions between the pragmatics of such systems (how they are used) and the semantics emerging in those systems (what the words, symbols, etc mean) is a fundamental issue that deserves greater attention. Equipping engineers of social machines with the right tools to achieve and maintain desirable system properties is a problem of practical relevance that needs to be addressed by future research.

WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.

Categories and Subject Descriptors

H.0 [**Information Systems**]: GENERAL

General Terms

Design, Theory

Keywords

social machines, software engineering

1. REFERENCES

[1] E. Yu. *Modelling Strategic Relationships for Process Reengineering*. PhD thesis, Department of Computer Science, University of Toronto, Toronto, Canada, 1995.

[2] A. Zubiaga, C. Körner, and M. Strohmaier. Tags vs shelves: from social tagging to social classification. *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, pages 93–102, 2011.