

A Universal Socio-technical Computing Machine

Markus Luczak-Roesch¹, Ramine Tinati¹, Saud Aljaloud¹, Wendy Hall¹ and Nigel Shadbolt²

¹ University of Southampton, Electronics and Computer Science, United Kingdom, mlr1m12@soton.ac.uk, <http://social-computer.org>

² University of Oxford, Department of Computer Science, United Kingdom

Abstract. This is an attempt to develop a universal socio-technical computing machine that captures and coordinates human input to let collective problem solving activities emerge on the Web without the need for an a priori composition of a dedicated task or human collective.

1 Introduction and Background

Since the advent of the *Social Machines* paradigm as abstractly described by Tim Berners-Lee [1], scholars have witnessed various attempts to underpin it with a formal theory and practice. The current range of theories includes a scheme to classify Web applications along a dedicated set of socio-technical properties [9, 10] as well as an archetypal framework to reflect upon sociality in collective action on the Web [11]. These two qualitative and small scale approaches are complemented by a quantitative information-centric view to Social Machines [5, 4, 6]. It turns out that, while these approaches provide novel means to retrospectively look at the interplay of the technical and the social on the Web, their constructive dimension – the practice – is somewhat limited.

Here we describe a novel architecture of a universal socio-technical computing machine – or short, a *Social Computer*. In contrast to the classification work, but in-line with ideas of archetypal narratives, our approach assumes Social Machines being the emergent output of human activity rather than any fixed engineered input. We ultimately seek to develop an engine that allows for actively shaping the morphology of the archetype of a collective action as it emerges in near real-time. Our approach differs from the typically coordinated approach in human computation and crowdsourcing [7, 3], where research commonly calls for methods to pre-engineer the way a human collective is going to perform a task [8]. Based on the principle that *the accumulated information sharing activities of individuals on the Web can compose purposeful collective action* [4], we designed a system that lets the human participants determine the computational program by their real-time inputs, while the technical components simply facilitate information flow to other technical systems to reach further human participants. Or to put it differently, **we developed a system that reacts upon bursts of information occurring on the Web and engages with human participants on various platforms to let a coordinated problem solving activity emerge.**

2 System Architecture

For the principled design of our system we rely on the representational state transfer (REST) principles of the Web architecture [2] and take a data-centric approach. That means that the central interface between system components is a data repository that is read from and written to by the individual components via RESTful request. The data in the repository is semi-structured so that the system allows for flexible expansion. Figure 1 depicts how the individual components of the system interact.

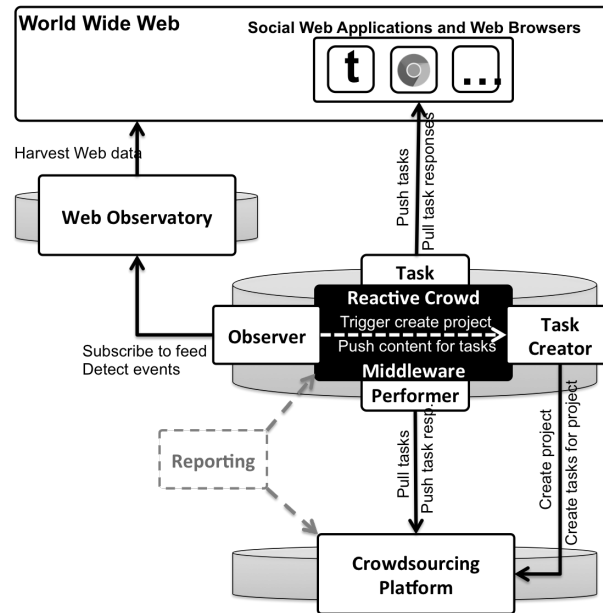


Fig. 1: Principled architecture of our universal socio-technical computing machine.

The source of data to be observed is any system on the World Wide Web. In order to decrease the necessary effort to implement individual data harvesters per system that can be accessed it is recommended to instantiate or link with a real-time Web Observatory [12], which is a decentralised approach to enable access to historic and real-time data from and about systems on the Web. The observer component subscribes to a unified activity feed on a Web Observatory and implements a) an information extraction mechanism to look for patterns in the content elements of the feed that are regarded relevant; and b) a threshold heuristic to indicate a relevant burst of activity around a particular pattern. When the heuristic indicates a burst, a project to manage crowdsourcing tasks on all incoming content containing the respective pattern is kicked off. The task creator manages that, from now on, any content element with that pattern is persisted in a crowdsourcing platform, which maintains a specific part of the

overall data repository focused on crowdsourcing analytics for making decisions about the completion of particular tasks and projects. The task performer component revisits the projects and tasks maintained in the data repository regularly and pushes them back to the Web to call for contributions from the crowd. This component also pulls responses to those published tasks back from the Web and persists them as task runs in the crowdsourcing database. This setup allows contributions to the crowdsourcing tasks in various ways: 1) participants of systems to which tasks are pushed (e.g. Twitter or facebook, see Figure 3) can simply reply to the shared content that contains the tasks (these are registered or unregistered participants, depending on the remote system’s policies); 2) participants can subscribe to the RESTful interface of the task performer and pick up pushed tasks and post task runs; 3) participants can log into the crowdsourcing platform that is part of this architecture and contribute to the tasks through the system’s Web interface. Our system is ultimately designed to allow for orchestrating arbitrary task workflows solely by the input from human users. In its current development state fixed workflow templates for translate, question answering and annotation tasks are provided.

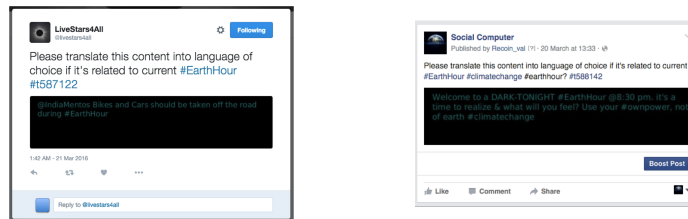


Fig. 3: Two ways to contribute human input to our Social Computer: Twitter and facebook.

3 Summary and Outlook

In this paper we presented a system design of a middleware to enable the autonomous creation and management of crowdsourcing tasks workflows. The approach automatically spins up crowdsourcing tasks for topics that feature temporally high activity on the Web in near real-time. We called this autonomous reactive crowdsourcing approach a *universal socio-technical computing machine* (or a Social Computer) because in its ultimate vision it shall allow for the composition of arbitrary workflows solely by the input of human participants through a set of primitive built-in tasks, which would form the basic instruction set of the Social Computer to form complex algorithms. This requires future work on the definition of this fundamental and generic instruction set and to implement it in a way that it can be used by the crowd in their responses to pushed out tasks

rigorously but still intuitively. Comprehensive experimentation is needed to understand the properties and impact of varying activity and consensus thresholds, and a completely new set of analytics to observe the system needs to be devised. Our Social Computer is configured by the content stream that is taken in and by the patterns to react upon within that stream, which allows for realizing either completely open or context specific work to be carried out. We see great potential for the approach to be used in scenarios that are inherently broadcasting orientated and do not feature a pre-defined online community to engage with. We find those in real-time event response such as disaster management using social media as well as in citizen science. The system also shows great potential to be used in organisations to let coordinated collaboration emerge when related activity around a topic is detected in independent organisational units.

References

1. T. Berners-Lee, M. Fischetti, and M. L. Foreword By-Dertouzos. *Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor*. HarperInformation, 2000.
2. R. T. Fielding. *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California, Irvine, 2000.
3. A. Kittur et al. The future of crowd work. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 1301–1318. ACM, 2013.
4. M. Luczak-Roesch, R. Tinati, K. O’Hara, and N. Shadbolt. Socio-technical computation. In *Proceedings of the 18th ACM Conference Companion on Computer Supported Cooperative Work & Social Computing*, pages 139–142, 2015.
5. M. Luczak-Roesch, R. Tinati, and N. Shadbolt. When resources collide: Towards a theory of coincidence in information spaces. In *Proceedings of the 24th International Conference on World Wide Web Companion*, pages 1137–1142, 2015.
6. M. Luczak-Roesch, R. Tinati, M. Van Kleek, and N. Shadbolt. From coincidence to purposeful flow? properties of transcendental information cascades. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, pages 633–638. ACM, 2015.
7. T. W. Malone, R. Laubacher, and C. Dellarocas. *Harnessing crowds: Mapping the genome of collective intelligence*. 2009.
8. P. Minder and A. Bernstein. Crowdlang: A programming language for the systematic exploration of human computation systems. In *Social Informatics*, pages 124–137. Springer, 2012.
9. N. R. Shadbolt et al. Towards a classification framework for social machines. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 905–912. International World Wide Web Conferences Steering Committee, 2013.
10. P. Smart, E. Simperl, and N. Shadbolt. A taxonomic framework for social machines. In *Social Collective Intelligence*, pages 51–85. Springer, 2014.
11. S. Tarte et al. Archetypal narratives in social machines: Approaching sociality through prosopography. 2015.
12. R. Tinati, X. Wang, T. Tiropanis, and W. Hall. Building a real-time web observatory. *Internet Computing, IEEE*, 19(6):36–45, 2015.