Abstract

Technology and global connectivity are driving economic growth through the world at an increasing pace. The field of Social Technical Congruence (STC; Cataldo et. al., 2006) has recently developed to understand the synergies between technical development and communication. Implementation of the STC perspective can guide the design of more innovative products, processes and services to stimulate the US economy.

Designing a product, process or service requires communication, collaboration, and a shared understanding of the problem at hand. Communication patterns of the individuals working on the design as well as dependencies between various sub-parts of the product being designed are intricately entwined and affect the overall quality of the final outcome.

STC has been successfully applied to the software development process to identify how much people should be communicating and who they should be communicating with. While the current STC research community looks primarily at software development, we envision that these general concepts and algorithms will have a positive impact on many design processes. The fundamental research question is, how can the social communication and task dependency networks between users be modeled and applied to improve the design of products, processes, and systems?
STC: LINKS BETWEEN SOCIAL SCIENCE AND TECHNOLOGY

Introduction

Researchers in Social Technical Congruence (STC; Cataldo et. al., 2006) have examined a wide variety of Open Source software projects to explore the fundamentals of STC. Open Source projects provide excellent test beds for data-mining the social and technical aspects of the community because of the openness of information that is available. The Open Source development model has grown significantly over the years from the academic world where users shared small pieces of source code to communities of developers creating fully fledged operating systems with huge corporate sponsorship (e.g., Red Hat Linux, Ubuntu Linux, etc.). Open Source was created with the idea that the source code of the program should be available and accessible to anyone. Allowing users to add features, fix bugs, and give suggestions have been powerful advantages to Open Source. The licenses of Open Source software require changes to be made available to the community at large. In addition, most Open Source software development is done through the Internet where members are often geographically dispersed around the world. Therefore, communication among all members then becomes critical for the group to function efficiently.

Most of the communication is done through email, mailing lists, forums, live chat, or other text-based communication services. Because of limited to no face-to-face or physical interaction, knowledge of other members is primarily reflected in the quality of their code submissions and their commitment to the project as a whole. Knowledge of the group members’ social customs, emotionality, actions, and goals (important indicators of group cohesion and effectiveness) are virtually nonexistent. On the other hand, this reliance on text-based communication levels the playing field for power and dominance in the group (Sproull and Kiesler, 1986). As a result, the members who are able to communicate effectively, to rally support for issues they feel are important, and to show their expertise, are able to influence the perception of the group. This results in a dynamic social power structure that is constantly evolving and changing.

In software development, the structure and dependencies of the source code as well as the relationships and communications between developers has been shown to be inextricably tied. The term STC, refers to the relationship between the code structure and communication structure (Cataldo et. al., 2006). Understanding that relationship aids in understanding a group’s productivity (i.e., task efficiency, overall time developing a project, etc.). In an initial demonstration in an Open Source context, Cataldo et al. used archival data to examine communication involved in handling software developers’ modification requests (e.g., request for a “bugfix”). Measures of coordination requirements (i.e., the code structure) and coordination activities (i.e., communication structure) were used to create a “congruence matrix”. As expected, when the congruence or “fit” between the task dependencies and the coordination activities of group members is high, then the project is healthier and has a shorter development time. In other words, high STC is related to a project group’s efficiency. With regard to the Open Source projects, sifting through large amounts of email to understand overall code changes as well as the design goals going forward takes time and diverts time from the overall project. Thus, developers need to be able to communicate quickly and effectively with the right people while at the same time preventing unnecessary communication.

In addition, the modularity of the code structure is one of the driving forces behind the need for STC in software development. For example, creating a sub-part of a project that stands on its own can be a powerful and efficient solution to coordinating large projects. As developers work to modularize
STC: LINKS BETWEEN SOCIAL SCIENCE AND TECHNOLOGY

the project as a whole, one consequence is that dependencies between modules are created. Some pieces of code may be crucial for the entire project while others may be less important. As developers make changes to the code base, they need to understand how these changes affect dependent modules as well as effectively communicate these changes to the correct developers. The same scenario can be applied to complex system design.

Researchers from the STC perspective have created a variety of metrics and algorithms for quantitatively determining gaps in project communication that could be hurting the health of the project (Sack et al., 2006). This recent yet growing research area is in its infancy and many of the algorithms need further enhancement to account for additional variables. For example, from the communication structure side, using STC to better understand how existing Open Source project members can effectively coordinate with and retain new members may lead to greater STC principles and ultimately greater efficiency within a given Open Source project. In addition, the current algorithms would benefit from taking into account temporal changes across the evolution of the project to match communication with specific changes in the software. We see the potential for this research to be generalized to work with any collaborative group designing a product.

Application to Overarching Design

In any design process, a group of people are communicating and collaborating in order to come up with a product, process, or system. In large groups, or when members are geographically dispersed, it can become difficult for individuals to understand the goals and actions of others to successfully tie in work they are doing with the group as a whole. Given the utility of the STC perspective in Open Source software design, we believe that more research is needed to better understand the interaction between social communication and task communication that goes on within a design group. Therefore, generalized algorithms and tools need to be developed to help users identify changes that have been made in the product or design, how these changes affect them, what communication is pertinent to their section or module, and how their module fits into the overall product or system design.

Successful completion of this work will require researchers from many different disciplines. For example, Artificial Intelligence researchers could examine the group’s communication style using natural language processing and speech systems to identify the topics of discussion and who was the group leader based on tone and inflection. Cognitive Psychologists and Sociologists could examine socialization processes of new and existing group members to better understand how communication is supported in the design process. Comparison of congruent modules and less congruent modules may lead to explication of the congruence process within the same larger project. Computer Scientists could tap into the available information and use data mining to determine whether a lack of communication means everyone is working hard and understands their roles or if people are confused and lack direction. Thus, all these fields will be important in furthering research in this area.

Technological change and innovation is implemented ultimately through social systems. The two are inextricably tied such that changes in one will affect the other. If researchers can identify ways to maximize congruence, it will benefit all systems to not only help in the design process but to create better products, processes and services for the end consumer.

Conclusion
In an increasingly interconnected world, both communication and technical innovation contribute to successful products, processes and systems. Companies and individuals need to explore methods that support both their technological advances and effective communication management systems to help them understand their social structure not only at their office or place of work but throughout the world. Research in the area of STC will help us better understand these connections and build algorithms to guide effective system design. This research will enable technological innovation and help us maintain our global leadership in technological advancement. The future of research is increasing collaboration between the social and technical sciences to create the next wave of intelligent systems that can leverage both human capital and technology advances.
References


License

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-sa/3.0/ or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.