Remixing Generic Defaults with Specialized Software: The Case of Enterprise Collaboration Systems

Emergent Research Forum (ERF)

Mazen Shawosh
University of Georgia
m.shawosh@uga.edu

Hani Safadi
University of Georgia
hanisaf@uga.edu

Nicholas Berente
University of Notre Dame
nberente@nd.edu

Abstract:

Extant literature emphasizes the fit between tasks and technology features to impact performance. Specialized information systems such as Enterprise Collaboration Systems (ECSs) are designed to align with the tasks of collaboration, coordination, and communication in organizations. Despite this fit, users often continue to use generic default technologies (i.e., general purpose software such as an Excel spreadsheet) to perform the tasks that are supported by the specialized systems. We investigate this phenomenon using a multiple case study approach. We present our initial findings and elaborate on the emerging theme of remixing of generic defaults with specialized software as a new perspective in exploring this phenomenon.

Keywords

Generic defaults, remixing, fit, task-technology fit, enterprise collaboration systems, ECS.

Introduction

The Enterprise Collaboration Systems (ECSs) (e.g., Groupware, Collaborative Product Commerce, Learning Management Systems) market is expected to grow up to$34.57 billion by the year 2023[1]. Many organizations have implemented some sort of ECS to facilitate business communications and information exchange between users. ECS technologies provide environments for sharing data/documents, facilitating collaboration and discussion, and enhancing work management. Nevertheless, even with the availability of such technologies that are well designed for these tasks, users combine *generic default software* (e.g., spreadsheets and personal emails) along with ECSs. These generic defaults are general technologies designed with generic features that are adaptable to different tasks.

The extant literature states that when misalignment exists between the tasks that users execute and the features (e.g., access, control, and operation) provided by an information system, users will behave in ways to resolve such misalignment, either by developing shadow systems and/or by improvising workarounds (Alter 2014; Soh et al. 2000). Although not always explicitly, the literature implies some form of task-technology misfit (Goodhue and Thompson 1995) as a key explanation for workarounds and shadow systems (e.g., Wei et al. (2005), and Soh and Sia (2004)).

Notably, these workarounds and shadow systems mainly appear with the implementation of Enterprise Resource Planning (ERP) systems in organizations [e.g., Boudreau & Robey (2005) and Berente & Yoo (2011)]. High chances of shadow systems and workarounds are linked with the implementation of ERPs

[1] Enterprise Collaboration Market - Global Forecast to 2023 - https://www.prnewswire.com/news-releases/enterprise-collaboration-market---global-forecast-to-2023-300649729.html

due to different "company-specific, public sector-specific, or country-specific requirements that did not match the capabilities of the ERP package" (Soh et al. 2000, p. 48). ERP systems span entire organizations and impose a particular way of organizing that fits some areas of an organization better than other areas. Thus there is inevitably some form of misfit in an ERP contexts.

It is puzzling, however, that even though ECSs are better aligned with coordination and collaboration tasks, users still deviate to using generic default systems. Thus, we seek to investigate why users use generic defaults instead of or along with better-fitting ECSs. In particular: *when there is a clear task-technology fit, why do users utilize generic defaults for performing tasks already supported by ECS?* We examine the usage patterns of ECS features and the tasks that are supposed to be associated with them, as well as how and why users utilize generic defaults in the same context while attending to systems' granularity. We present our initial findings and discuss an interesting emerging theme in the data.

Literature Review

Enterprise Collaboration Systems (ECSs) contain many features that facilitate active collaboration and communication among users. Unlike ERP systems, ECSs are well suited to the tasks of collaboration and communication. According to Greeven and Williams (2016), "ECS combine Enterprise Social Software (ESS) components (e.g. social profiles, tags, wikis, blogs) with traditional groupware components (e.g. e-mail, group calendars, document libraries) to support social business and communication, collaboration, content and knowledge sharing within organizations" (p. 179). This class of information systems appears under different names. For instance, Collaborative Product Commerce (CPC) is software used by product design engineers to facilitate communication and management of data used in product design and development (Banker et al. 2006). Groupware is another software that combines tools for "document sharing and review, concurrent editing, and shared calendars" (Lanubile et al. 2010, p. 53). A third type of ECS is Learning Management Systems (LMS), which provide "an array of tools and functions to support teaching and learning, usually including course management tools, online group chat and discussion, homework collections and grading, and course evaluation" (Yueh and Hsu 2008, p. 60).

It is important to establish the difference between ECS and ERP systems. In one key dimension, ECS are designed to augment joint work among users, while ERP systems are process-oriented and designed with the intention to support business functions and processes. Other dimensions include how these two classes of information systems are implemented and used in organizations (Schubert and Glitsch 2015). It is not unusual to observe that ERP system use is mandatory by organizations, whereas the use of ECS is voluntary, although recommended.

Methodology

To conduct this study, we will closely follow Eisenhardt's (1989) process for conducting qualitative studies. The main reason for this choice is that the guidelines provided combine the best-of-breed inductive research steps and activities mentioned in the past literature. Initially, the study was designed to investigate our first thinking of the research problem, which is understanding the misfits in ECS and the use of generic defaults for tasks execution. We entered the research field with knowledge of prior research such as Task-Technology fit (TTF), and shadow systems and workarounds literature (Alter 2014; Chua et al. 2014; Ferneley and Sobreperez 2006; Wei et al. 2005).

For data collection, we primarily depend on interviewing instructors of record (i.e., users) at a state university in the United States. We conducted the interviews with users primarily from one academic department. We developed an interview guide with semi-structured questions with five sections investigating the idea of "misfit" by asking questions related to fit dimensions such as currency, systems reliability, and the right level of data (Goodhue and Thompson 1995). We interviewed ten instructors of record, with an average of 35 minutes per interview. The researcher who conducted the interview is part of the same department as the participants, except for one interviewee from another department in the same college. We interviewed eight faculty members (four tenure-track and four non-tenure track) and two Ph.D. students. All interviews were recorded and transcribed with the permission of the participants. Extensive notes were taken during the interviews.

The academic institution implemented a learning management system (hereinafter called LMS) more than five years ago. Using the system is not mandatory. However, instructors are highly encouraged to utilize it for course management. This study takes place while the system is fully functional and supported across the institution. Table 1 depicts the steps involved in the study.

Step	Activity
Getting started	Research Question: • When there is a clear task-technology fit, why do users utilize generic defaults for performing tasks already supported by ECS? A priori constructs: Task-Technology fit. Generic vs. specialized technology.
Selecting cases	Specify population: • University instructors of record. Theoretical sampling: • Will be performed after the first round of analysis.
Instrument	Mainly interviews. Data triangulation from sources such as surveys and documentations could be used during the study.
Overlap between data collection and analysis	The interview guide is updated after some interviews to accommodate more interesting thoughts.
Analyzing data	Compare and abstract from different interviews. Notes and transcription of interviews. Different coding techniques (Corbin and Strauss 2014; Saldana 2015).
Shaping hypotheses	Although users perceive a general fit between ECS and their job tasks, they deviate to using generic ad hoc defaults for performing tasks already supported by ECS.
Enfolding literature	Comparing with conflicting and similar literature: • TTF, workarounds, and shadow systems.
Reaching closure	When we reach a point of minimal incremental learning about the phenomenon.

Table 1. Steps for designing the research and analyzing data

Initial Findings

The study is still underway. Here, we present our initial findings based on preliminary analysis of the first phase of data collection and discussions among the authors.

Teaching is a dynamic job, and each course has its own requirements. Using LMS alone to manage courses would inhibit both instructors and students from obtaining the most benefits from the courses. The LMS system provides features designed to support teaching tasks. These features are intentionally and specifically designed to support coordinating and managing classes and courses. What we observed, however, is that in addition to using LMS features, instructors also deviate to using generic defaults and social software applications that are not built for direct support of teaching tasks. These tools are utilized for tasks ranging from simple manipulations of students' grades to executing complete tasks such as one-on-one communications and content/material sharing. Even those who are heavy users of LMS deviate to using such software when they perceive a subjective fit between the task and the software. The subjective (perceived) fit is a judgment of congruence for selecting the better choice (i.e., software) to execute a specific task at a specific time. To observe this subjective fit more clearly, the level of analysis should not be the software. During the interviews, participants talked more about LMS features than they did about LMS as a whole. Table 2 shows examples of different software used for teaching tasks.

One participant (P1) who uses several LMS features mentioned using Slack as a tool for group work coordination and collaboration instead of the functionalities available in LMS. This participant stated that "[LMS does] *have a discussion environment, but I'm not very happy with it in the sense that it's not very conducive to actually facilitating a discussion*." Also, the participant added that Slack " [is] *a nice way for project teams to work together, and organizations to collaborate on like activities and things of that nature*." Another participant (P2) stated using generic defaults for tasks like grade calculation. The participant stated "[I] *download the grades once in a while from* [LMS] *to Excel. So, I keep not only a*

back-up, but when I want to do some manipulation and some data crunching, I prefer to download it in Excel. I do it from there."

Type of Software	Research Site Examples
Specialized software (LMS)	Grade calculation. Communication with students. Content and material sharing. Discussions and collaborations.
Generic default software	Spreadsheets for grade manipulation and calculation. Self-developed websites for content and material sharing. Official/personal email for communication.
Social software	Google Hangouts and Skype for virtual meetings and communication. Slack for team projects and collaboration.

Table 2. Examples of the different software used on the research site

Emergent Theme: Remixing

We define remixing as *the act of combining specialized software (e.g., LMS) with generic defaults to accomplish an ongoing set of small tasks around broader tasks*. For example, communication is a broad task that involves smaller tasks of creating class announcements, one-on-one email exchanges, and student groups' communications. The theme of remixing emerged during interviews when more probing questions were asked to investigate the reasons for combining generic defaults with LMS to perform teaching tasks.

Misfits in ERP systems are either deficiencies or impositions (Strong and Volkoff 2010). Is this the same case in ECS? What surfaced from the interviews is that ECS on the research site (i.e., LMS) does not suffer from a lack of features. According to several participants, there are more functionalities available than are actually used. Nevertheless, participants still extended generic defaults when executing teaching tasks. When evaluating fit, TTF focuses on one technology at a time and assumes that users are rational agents (Dishaw et al. 2002). A user faces a task and considers whether certain features in the technology would accomplish such a task. When there is an alignment between the task and the features, the user performs the task, experiences positive performance impact (Goodhue and Thompson 1995; Mathieson and Keil 1998), and moves to another task. However, the real world functions in a more complex fashion.

The fit view is atomistic and oversees the reality of tasks and technologies. A university professor, for example, does not perform one task at a time. She might be teaching classes, performing administrative work, conducting research, and consulting, and she assumes different and concurrent roles and responsibilities. Some tasks are related, and others are not. To perform these tasks, there are several technologies the professor may have already selected, as well as technologies that she is accustomed to selecting. In other words, there is an ongoing flux and flow of related and unrelated tasks that are performed using different software applications.

Here, the use of generic defaults emerges. The professor is familiar with generic defaults that can be extended to related and unrelated tasks. Such generic defaults help her navigate the complex flow of tasks and technologies. Thus, economizing and achieving efficiency in performing a single task is merely a small part of a bigger picture. An individual chooses to remix generic defaults with ECS because the goal is achieving overall efficiency in performing multiple tasks.

Conclusion

The next step in our study is to collect more data about the emerging concept of remixing. We will use the theoretical sampling method to develop more understanding about the dimensions and properties of remixing with generic defaults. We aim to collect additional data about possible remixing patterns around the tasks performed by participants.

We anticipate that our research will contribute to further understanding the paradox of ECS fit. We will try to go beyond the theories of TTF and workarounds to shed more light on the phenomenon. We also

anticipate that our research will have practical implications for organizations that heavily invest in ECSs and expect employees to use them.

REFERENCES

Alter, S. 2014. "Theory of Workarounds," *Communications of the Association for Information Systems* (34:1). (https://doi.org/10.17705/1CAIS.03455).

Banker, R. D., Bardhan, I., and Asdemir, O. 2006. "Understanding the Impact of Collaboration Software on Product Design and Development," *Information Systems Research* (17:4), pp. 352–373. (https://doi.org/10.1287/isre.1060.0104).

Berente, N., and Yoo, Y. 2011. "Institutional Contradictions and Loose Coupling: Postimplementation of NASA's Enterprise Information System," *Information Systems Research* (23:2), pp. 376–396. (https://doi.org/10.1287/isre.1110.0373).

Boudreau, M.-C., and Robey, D. 2005. "Enacting Integrated Information Technology: A Human Agency Perspective," *Organization Science* (16:1), pp. 3–18. (https://doi.org/10.1287/orsc.1040.0103).

Chua, C., Storey, V., and Chen, L. 2014. "Central IT or Shadow IT? Factors Shaping Users' Decision to Go Rogue With IT," in *ICIS 2014 Proceedings*, , December 15.

Corbin, J., and Strauss, A. 2014. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, (Fourth edition.), Los Angeles: SAGE Publications, Inc.

Dishaw, M., Strong, D., and Bandy, D. B. 2002. "Extending the Task-Technology Fit Model with Self-Efficacy Constructs," in *AMCIS 2002 Proceedings*, , December 31.

Eisenhardt, K. M. 1989. "Building Theories from Case Study Research," *Academy of Management Review* (14:4), pp. 532–550. (https://doi.org/10.5465/amr.1989.4308385).

Ferneley, E. H., and Sobreperez, P. 2006. "Resist, Comply or Workaround? An Examination of Different Facets of User Engagement with Information Systems," *European Journal of Information Systems* (15:4), pp. 345–356. (https://doi.org/10.1057/palgrave.ejis.3000629).

Goodhue, D. L., and Thompson, R. L. 1995. "Task-Technology Fit and Individual Performance," *MIS Quarterly* (19:2), pp. 213–236. (https://doi.org/10.2307/249689).

Greeven, C. S., and Williams, S. P. 2016. "Enterprise Collaboration Systems: An Analysis and Classification of Adoption Challenges," *Procedia Computer Science* (100), International Conference on ENTERprise Information Systems/International Conference on Project MANagement/International Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN / HCist 2016, pp. 179–187. (https://doi.org/10.1016/j.procs.2016.09.139).

Lanubile, F., Ebert, C., Prikladnicki, R., and Vizcaíno, A. 2010. "Collaboration Tools for Global Software Engineering," *IEEE Software* (27:2), pp. 52–55. (https://doi.org/10.1109/MS.2010.39).

Mathieson, K., and Keil, M. 1998. "Beyond the Interface: Ease of Use and Task/Technology Fit," *Information & Management* (34:4), pp. 221–230. (https://doi.org/10.1016/S0378-7206(98)00058-5).

Saldana, J. M. 2015. *The Coding Manual for Qualitative Researchers Third Edition*, (3rd edition.), Los Angeles ; London: SAGE Publications Ltd.

Schubert, P., and Glitsch, J. H. 2015. "Adding Structure to Enterprise Collaboration Systems: Identification of Use Cases and Collaboration Scenarios," *Procedia Computer Science* (64), Conference on ENTERprise Information Systems/International Conference on Project MANagement/Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN / HCist 2015 October 7-9, 2015, pp. 161–169. (https://doi.org/10.1016/j.procs.2015.08.477).

Soh, C., Kien, S. S., and Tay-Yap, J. 2000. "Enterprise Resource Planning: Cultural Fits and Misfits: Is ERP a Universal Solution?," *Communications of the ACM* (43:4), pp. 47–51. (https://doi.org/10.1145/332051.332070).

Soh, C., and Sia, S. K. 2004. "An Institutional Perspective on Sources of ERP Package–Organisation Misalignments," *The Journal of Strategic Information Systems* (13:4), Special issue "Understanding the Contextual Influences on Enterprise System Design, Implementation, Use and Evaluation," pp. 375–397. (https://doi.org/10.1016/j.jsis.2004.11.001).

Strong, D. M., and Volkoff, O. 2010. "Understanding Organization—Enterprise System Fit: A Path to Theorizing the Information Technology Artifact," *MIS Quarterly* (34:4), pp. 731–756. (https://doi.org/10.2307/25750703).

Wei, H.-L., Wang, E. T. G., and Ju, P.-H. 2005. "Understanding Misalignment and Cascading Change of ERP Implementation: A Stage View of Process Analysis," *European Journal of Information Systems* (14:4), pp. 324–334. (https://doi.org/10.1057/palgrave.ejis.3000547).

Yueh, H.-P., and Hsu, S. 2008. "Designing a Learning Management System to Support Instruction," *Communications of the ACM* (51:4), p. 59. (https://doi.org/10.1145/1330311.1330324).