Digital Infrastructure Evolution: A Digital Trace Data Study

Katharina Drechsler
University of Liechtenstein, katharina.drechsler@uni.li

Thomas Grisold
University of Liechtenstein, thomas.grisold@uni.li

Michael Gau
University of Liechtenstein, michael.gau@uni.li

Stefan Seidel
University of Liechtenstein, stefan.seidel@uni.li

Digital Infrastructure Evolution: A Digital Trace Data Study

Short Paper

Katharina Drechsler
University of Liechtenstein
Vaduz, Liechtenstein
katharina.drechsler@uni.li

Thomas Grisold
University of Liechtenstein
Vaduz, Liechtenstein
thomas.grisold@uni.li

Michael Gau
University of Liechtenstein
Vaduz, Liechtenstein
Karlsruhe Institute of Technology
Karlsruhe, Germany
michael.gau@uni.li

Stefan Seidel
University of Liechtenstein
Vaduz, Liechtenstein
stefan.seidel@uni.li

Abstract

Digital infrastructures are socio-technical arrangements of physical objects, digital technologies, users, and processes. They constantly evolve and provide the foundation for the emergence and implementation of a variety of applications. But how and why do digital infrastructures evolve? In this short paper, we report on an ongoing study of an Industrial Internet of Things (IIoT) system by drawing on digital trace data about the system's development process and using code complexity analysis. We present interaction patterns between the development of a digital infrastructure's core services on the one hand and the development of specific applications on the other. Based on these patterns, we point to several puzzles that we identify around digital infrastructure evolution.

Keywords: Digital infrastructure, digital trace data, computationally intensive theory construction, industrial internet of things, system development

Introduction

Digital infrastructures have gained increasing attention in information systems research (e.g., Henfridsson and Bygstad 2013; Koutsikouri et al. 2018; Tilson et al. 2010a). They are socio-technical arrangements of physical objects, digital technologies, users, and processes. Digital infrastructures continuously evolve (Tilson et al. 2010a). As more applications (e.g., services) are added to a digital infrastructure, it exhibits new features and value propositions and grows into more complex forms over time (Koutsikouri et al. 2018). Consider the Internet, which is the most prominent digital infrastructure, and which generates an ever-increasing surge of new features and services as more applications are being added (Hanseth and Lyytinen 2010).

Previous research has provided a variety of perspectives on the evolution of digital infrastructures (e.g., Hanseth and Modol 2021; Henfridsson and Bygstad 2013; Koutsikouri et al. 2018). Across these works, we can recognize a strong focus on the social, organizational, and structural dynamics that occur around digital infrastructure evolution. For instance, studies have focused on tensions that arise between infrastructure providers, developers, and users (Osmundsen and Bygstad 2022). What has received little attention, however, is how infrastructure evolution unfolds at the technical level and how changes at the infrastructural level interact with changes at the level of specific applications. Understanding the material evolution of digital infrastructures is, however, important if we want to embrace digital infrastructures as social *and* technical systems (Sarker et al. 2019; Tilson et al. 2010a).

This short paper reports on an ongoing research project where we study the evolution of an Industrial Internet of Things (IIoT)-based system from a software development perspective. IIoT-based systems "connect[ing] all the industrial assets, including machines and control systems, with the information systems and the business processes" (Sisinni et al. 2018, p. 2) and are a prime example of digital infrastructures—they are socio-technical networks of physical objects and digital technologies operating to achieve an organization's goals (Tilson et al. 2010a). IIoT systems (hereafter, we refer to them as IIoT infrastructures) have key characteristics of digital infrastructures (Osmundsen and Bygstad 2022; Tilson et al. 2010a); they are layered, scalable, and flexible, and components—such as core services and applications—recursively build on each other.

Engaging in computational theory construction (Berente et al. 2019; Miranda et al. 2022), we explore the evolution of the IIoT infrastructure at a manufacturing company by analyzing digital trace data of the system's software development. These digital trace data contain developers' activities as they gradually extended, changed, and refined the digital infrastructure by (1) adding *IIoT infrastructure services* and (2) creating *IIoT applications*. *IIoT infrastructure services* were created by connecting hardware through software and establishing cloud-hosted services, for instance, to enable machine connectivity. This provided the foundation for *IIoT applications* or specific "use cases" that capitalize on the IIoT infrastructure for collecting and analyzing data generated by a particular production machine or manufacturing process.

More specifically, we collected and analyzed digital trace data from three key repositories: two software development repositories that reflected activities pertaining to the software development of the IIoT infrastructure's core services and applications, and one issue tracking repository that reflected software development tasks and hardware requests of specific applications in the IIoT infrastructure. We present preliminary findings based on the application of code complexity analysis that highlight the dynamics of how an IIoT infrastructure grows into more complex forms over time. Specifically, we identify interaction patterns between the development of digital infrastructure's core services and the development of specific applications and their related use cases. We provide tentative explanations about how one level affects the other. By sketching out future research activities that we plan to pursue, we highlight how digital trace data offer ample opportunity to study the co-evolution of digital infrastructures and specific applications.

In the following, we provide a brief overview of existing research regarding digital infrastructure evolution. We then outline our case setting, methodological approach, and the digital trace data we analyzed. We present preliminary findings and identify puzzles emerging from our analysis. Building on these, we briefly discuss our findings and outline our next steps in this research project.

Theoretical Background: Digital Infrastructure Evolution

Digital infrastructures are shared, open, and unbounded socio-technical systems connecting heterogeneous sets of IT capabilities, users, and operations (Hanseth and Lyytinen 2010). They are a key type of socio-technical systems (Tilson et al. 2010a). Due to their heterogeneity and complexity, digital infrastructures cannot be defined in terms of their functionalities (Henfridsson and Bygstad 2013). To some extent, a digital infrastructure has a stable component that allows for adding new applications (processes and products). This stable component is characterized by centralized control that provides standards and services through which new applications can be connected (Koutsikouri et al. 2018). At the same time, as new applications are added, the digital infrastructure evolves (Modol and Eaton 2021) in response to demands from internal and external environments (Henfridsson and Bygstad 2013). Therefore, when we look at the functionalities of a digital infrastructure, we only gain a snapshot of what it does at a given point in time—the digital infrastructure itself is inherently incomplete, underspecified, and open for further developments (Tilson et al. 2010b).

Past research has studied how digital infrastructures evolve over time (e.g., Hanseth and Lyytinen 2010; Henfridsson and Bygstad 2013; Koutsikouri et al. 2018; Tilson et al. 2010b). Such research has foregrounded different aspects, such as strategies and tactics to enable digital infrastructure evolution (Koutsikouri et al. 2018), tensions that occur around digital infrastructure evolution (Montealegre et al.

2019), or mechanisms that underlie digital infrastructure evolution (Henfridsson and Bygstad 2013). Overall, previous research has emphasized social, organizational, and structural dynamics that emerge when managers, users, and developers engage in "negotiation, tensions, and conflicts" (Bygstad and Øvrelid 2020, p. 222) during infrastructure evolution.

What has received little attention, in contrast, is the material element of digital infrastructure evolution and how it is involved in those phenomena. After all, digital infrastructures involve layered and interconnected systems that are based on software standards and services (Bygstad and Øvrelid 2020; Koutsikouri et al. 2018). To this end, it has been noted, for example, that standards lead to path dependency because they enable and constrain what developers can and cannot do (Hanseth and Lyytinen 2010). In general, however, we know little about the patterns, mechanisms, and dynamics that unfold at the level of software code. Such insights are, however, important if we want to understand the evolution of digital infrastructures as socio-technical systems, where social and technical aspects interact as digital infrastructures grow into more complex forms over time.

Research Context

Case Setting

The case organization is a leading European manufacturing company, which operates worldwide. The company has implemented an IIoT infrastructure to address information needs in its production facilities. The IIoT infrastructure uses a centralized cloud-based IoT platform, consisting of a database and functionality to store, process, and analyze data. The IoT platform is connected to production machinery, sensors, and devices allowing real-time data analysis. It also integrates IIoT applications and applications for visualization and control (integrating data from infrastructure operations, including central and added services and products).

The case company developed and implemented the IIoT infrastructure using an incremental approach. After experimenting with small-scale IIoT solutions for specific product lines, management decided to support the implementation of a large-scale initiative to promote the extension and scaling of the IIoT infrastructure. Besides implementing a cloud-based IoT platform to centralize data processing and analysis, employees were encouraged to explore opportunities for digitalizing production processes through developing and implementing IIoT use cases. As a result, the IIoT infrastructure evolved continuously as newly identified information requirements led to investments in sensors and other hardware and the development of new software components.

Computationally Intensive Theorizing with Digital Trace Data

To trace IIoT infrastructure evolution, we use an abductive research approach that uses both human and machine pattern recognition (Lindberg 2020). While the evolution of digital infrastructure has been studied from various perspectives (e.g., Hanseth and Modol 2021; Henfridsson and Bygstad 2013; Koutsikouri et al. 2018), the material aspects of digital infrastructure evolution have received little attention. To gain insights into the evolution of digital infrastructure at the technical level, we thus choose to approach the research topic with an exploratory research method with the aim to generate theoretical insight. To this end, we conduct a computationally intensive analysis using digital trace data (Berente et al. 2019; Miranda et al. 2022). Digital traces provide potentially novel perspectives on socio-technical phenomena as they reflect activities and events that are generated as actors use digital technologies. Hence, such data represent "digital footprints" of actors, including software developers and their activities, who perform activities (Golder and Macy 2014).

Using digital traces to study digital infrastructure evolution provides two key advantages. First, digital trace data appear in large quantities and over extended periods of time (Lazer et al. 2020); thus, they provide fine-granular yet far-reaching insights into digital infrastructure evolution (Østerlund et al. 2020). Second, digital trace data include temporal information indicating when a specific activity was carried out (Pentland et al. 2020). Hence, they are particularly useful for visualizing and analyzing the dynamics that unfold around digital infrastructure evolution. Taken together, using digital trace data is particularly useful to reveal how and why a digital infrastructure evolves as it is being shaped by the activities of developers (Lindberg et al. 2016).

To analyze the trace data, we applied analysis techniques at the software code level to get insights into the evolution of the digital infrastructure. Code complexity analysis is a well-known and widely used method in software engineering (Antinyan et al. 2017). Code complexity is "the degree to which a system or component has a design or implementation that is difficult to understand and verify" (Geraci et al. 1991, p. 45). Furthermore, code complexity can measure the resources spent on developing, or maintaining, a solution for a given task (Basili 1980; Fenton and Bieman 2014). Many different measures exist to calculate code complexity, for instance, code size or code changes.

We applied the Lines of Code (LOC) measure (Boehm et al. 2000) to analyze code complexity and how it evolves over time. LOC is a measure to quantify code complexity by counting each line of software code in a single source file or an entire repository. LOC measures the size of a computer program and can be used to illustrate changes in size and complexity during development. Additionally, we used the number of activities such as commits and other related activities (e.g., merge or pull requests). Commits represent change activities across the project at given points in time. They thus reflect the extent to which changes are made to software code as it is developed and improved over time.

The digital trace data at our disposal record the company's first moves in establishing an IIoT infrastructure from autumn 2019 to December 2021, when several IIoT use cases were implemented to solve specific information requirements. We included three repositories in our analysis. First, we used a Git[1] repository that enabled us to analyze development activities pertaining to the cloud-based IoT platform and the infrastructure core services provided by that system. Second, we used another Git repository that recorded the software development activities related to the IIoT applications over time and allowed us to explore the creation of applications associated with specific use cases built upon those infrastructure core services. Third, we used a Jira[2] repository, which tracked issues describing software development tasks and hardware requests related to IIoT infrastructure services and applications, such as tracing a particular machine's temperature and other parameters during production or visualizing the overall equipment efficiency. Table 1 provides an overview.

Type of Repository	**Description**		
Software Development Repository (Git)	Software development activities contributing to the IIoT infrastructure evolution		
	Number of Actors	Number of Activities	Lines of Code
IIoT infrastructure services	6	931	17,384
IIoT applications	8	917	64,544
Issue Tracking Repository (Jira)	Issues describing software development tasks and hardware requests of IIoT infrastructure services and applications		

Table 1. Overview of the Data Sources

For our analysis, we sliced the data in the two Git repositories into subsets on a monthly basis. We calculated the frequencies of overall activities and the added LOC of the corresponding month, thus attending to the project's temporal development. Additionally, we used the issue tracking repository (Jira) to identify specific use cases within the Git repositories. To this end, we manually derived keywords from the Jira repository to describe three exemplary use cases. We then used these keywords to search for activities in the two Git repositories relating to these use cases. Finally, after observing that the overall activities peaked several times throughout the studied timeline, we extracted the commit messages for these activities. We coded these commit messages to gain a broad understanding of the main activities explaining these peaks in overall activities. We thus combined computational and manual analyses.

[1] Git is a version control software for source code.
[2] Jira is an issue tracking software.

Note that the digital trace data we analyze in this paper are just a subset of the overall data set we have been collecting. At the case organization, we also collected qualitative data through interviews and informal conversations. This data, for instance, helped us gain a general understanding of the research context and the use cases.

Preliminary Findings

In the following, we first describe the preliminary findings of our digital trace data analysis. Building on these insights, we then identify key puzzles that these findings offer and that we are planning to further investigate.

Figure 1 highlights the connection between infrastructure core service development and application development in terms of code complexity. Thus, we compare the number of LOC added to establish IIoT infrastructure services (green) and LOC added to establish IIoT applications (blue). Figure 1 indicates that software development of infrastructure core services occurred throughout the project but was particularly high at the beginning of the project (see, for example, the spike in June 2020). This indicates that the first phase of digital infrastructure evolution was characterized by a rapid extension of the underlying infrastructure services. Later, infrastructure services development continued alongside the development of applications. In contrast, no application development took place in the earlier stages of the IIoT infrastructure evolution. Then, in October 2020, added LOC of application development rapidly increased.

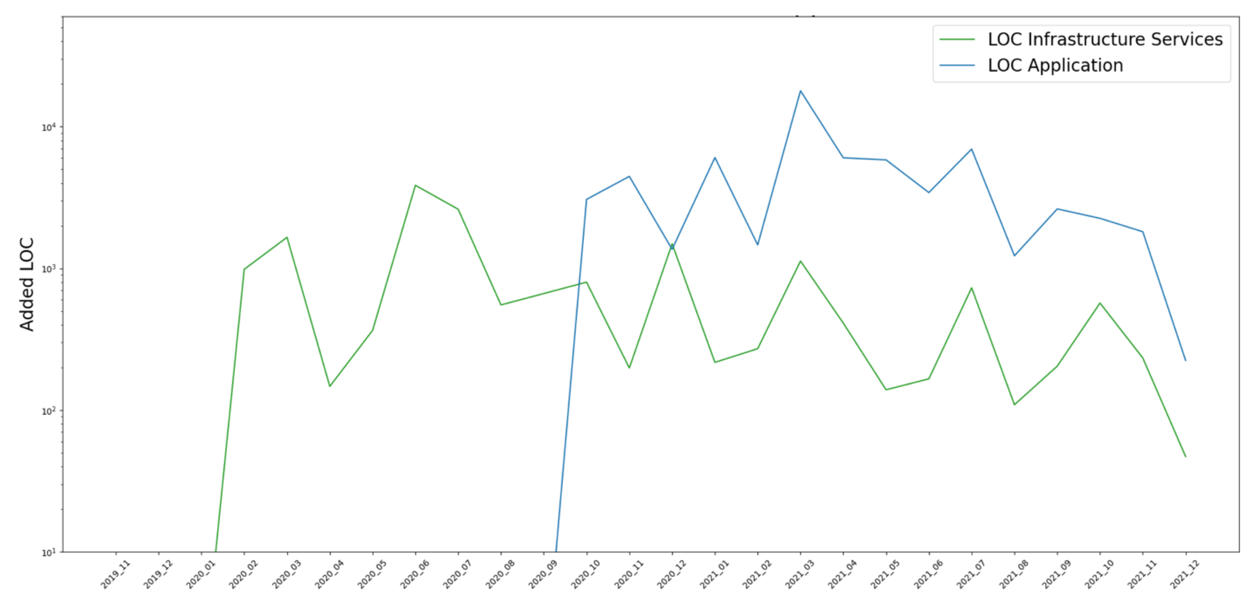

Figure 1: Code Complexity of the Infrastructure Services and the Application Repository Over Time

Legend: green = added LOC of infrastructure development, blue = added LOC of application development

This simple analysis suggests that application development occurs when the infrastructure services have been set up to a considerable extent. This reflects that, before a digital infrastructure can grow, the architecture needs to be developed, for example, in terms of software standards and services (Bygstad and Øvrelid 2020). When the infrastructure services take shape, application development follows. Thus, spikes in the added LOC of infrastructure services and applications co-occur over time. For instance, a spike in infrastructure services development in December 2020 was followed by a spike in application development around January 2021, and spikes co-occurred in March 2021 and July 2021. Application development makes use of digital infrastructure services.

We identify our first puzzle considering the added LOC produced at the infrastructure services and application development levels, as shown in Figure 1. Despite the similarities in the patterns of software development across the two levels, the pattern of correlation varies over time; a spike in the LOC of IIoT infrastructure services is not necessarily followed by a similar spike in LOC of IIoT applications. Moreover, the continuous software development related to IIoT infrastructure services is interesting. Even after establishing the most important core services during the first months of the IIoT infrastructure evolution, the software development activities surrounding IIoT infrastructure services remained high. This suggests that adding applications is related to extending infrastructure core functionality. So far, the reasons and mechanisms underlying the diverging patterns of correlation between application development and the development of infrastructure services are unclear. Different infrastructure services may offer varying levels of opportunity to develop applications. Some applications may demand new core services while others may not. Accordingly, we identify the following puzzle:

> ***Puzzle 1:*** *Application development and the development of infrastructure services are characterized by varying degrees of correlation across time.*

We now turn toward analyzing the development of specific use cases building on the IIoT infrastructure and its software applications. At the case organization, IIoT-based use cases are initialized and developed by employees to explore the IIoT infrastructures' opportunities. Figure 2 shows the relation between overall project activities (red—i.e., related to both infrastructure and application development) and the implementation of three specific use cases that were designed and implemented early in the project (orange, green and purple). We can see that use case development starts around May 2020. We observe that some spikes in overall software development activities co-occur with the development of the specific use cases. This could indicate that the development of larger use cases (e.g., use case 1 in Figure 2) drives overall infrastructure development.

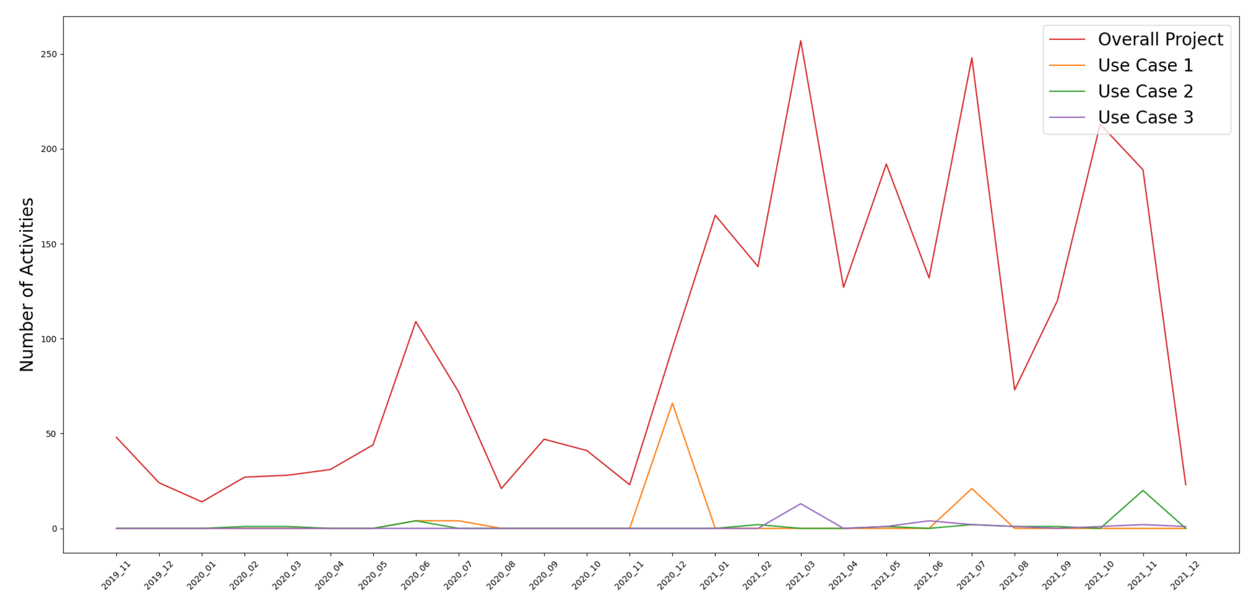

Figure 2: Activities Within Three Use Cases Compared to the Overall Project Activities

Legend: red = overall project activity across repositories; orange, green and purple = activities of use case 1, 2, and 3, respectively

Figure 2 also highlights that the activities surrounding the use case development only account for a small proportion of the overall project activities. Potential explanations that warrant further exploration include infrastructure services refactoring or a centralization of use case functionalities to make them available for other applications. Accordingly, we identify the following puzzle:

> ***Puzzle 2:*** *Use case development correlates with overall project activities but only explains a small proportion of them.*

Finally, our data extend insights into the relationship between centralization and decentralization in digital infrastructure evolution. Figure 3 depicts the implementation of the three different use cases (orange, green, and purple) at the software application level compared to overall project activities (red) and activities related to infrastructure services (grey). We can observe that after the initial infrastructure services are established (around August 2020), peaks in both overall development and infrastructure development tend to co-occur with peaks in use case development. This points to a process of centralization as functionality is moved from applications to infrastructure core services. At the same time, these software development activities vary depending on the specific use case (some use cases come with higher levels of infrastructure development than others). However, Figure 3—as well as our manual analysis of the Jira data—also indicates that, in some situations, functionality was moved to applications (see "Decentralization of Infrastructure Functions" in Figure 3). This finding offers another puzzle when considered within the broader context of IIoT infrastructure evolution:

> ***Puzzle 3****: Use case and application development may involve centralization and decentralization of functionality with varying degrees of changes in the level of infrastructure services required.*

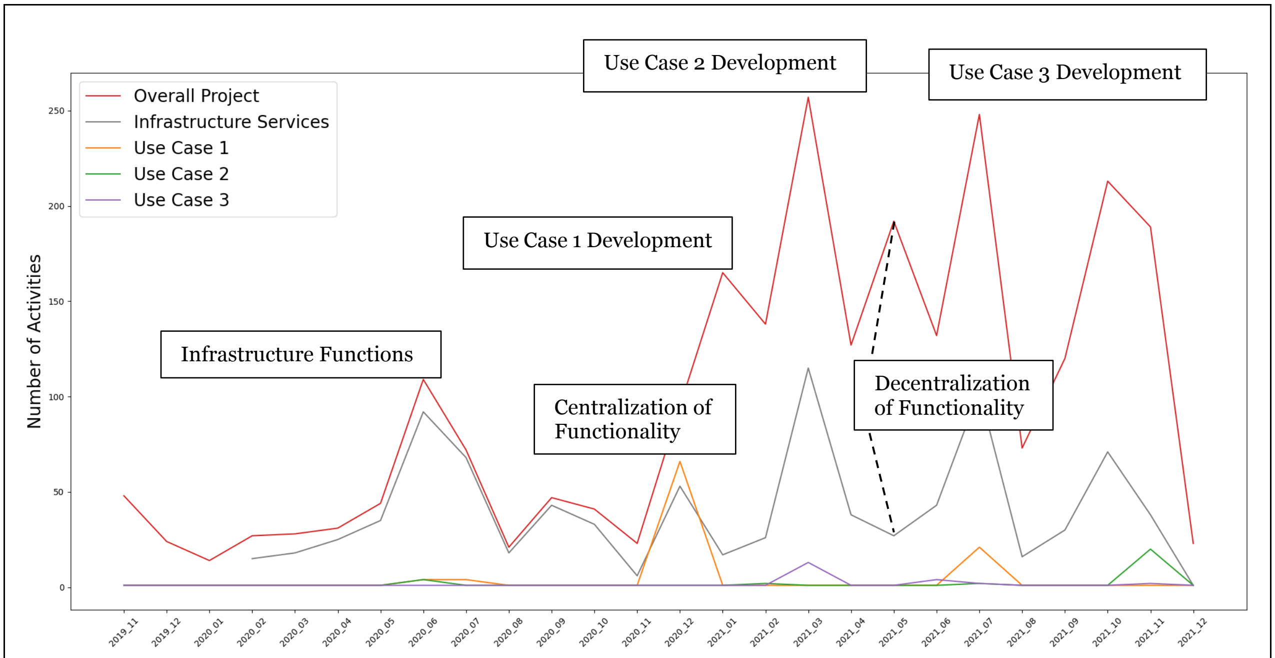

Figure 3: Main Topics of the Software Development Activities over Time

Legend: red = overall project activities across repositories; grey = activities of infrastructure services repository; yellow, green, and purple = activities of use case 1, 2, and 3, respectively

Discussion and Outlook

This short paper offers first insights into an ongoing research project where we use digital trace data to study the evolution of an IIoT infrastructure from the perspective of software development. While some of the general patterns identified (infrastructure services first, applications second) may not seem surprising, our analysis presents interesting puzzles surrounding the relationship between software development of infrastructure services, applications, and use cases on the one hand and the overall project on the other.

Our data suggest that stability precedes emergence: As soon as the infrastructure services are in place, applications are added to this infrastructure, leveraging the infrastructure's software standards and services. Changes to the infrastructure services occur to a lesser degree at later stages in the project—but they still occur. Yet, we see a substantial variance in the degree to which the development of applications and infrastructure services occurs. Thus, it will be interesting to see how specifically the infrastructure services are changed as applications are added. Additionally, we find that the development of infrastructure services and use cases correlates with overall project activities with highly varying degrees. Yet, relying on the current analysis, we fall short of offering and validating explanations underlying these patterns.

In moving forward, we will extend our analysis in two ways. First, we will broaden our computational analysis by considering the digital trace data from various perspectives. We plan to explore software development activities related to IIoT infrastructure services and applications using alternative measures of code complexity (e.g., cyclomatic complexity, Halstead volume) to gain a comprehensive understanding of IIoT infrastructure development. In the next step, we will use more sophisticated methods (e.g., topic modeling and text analysis) to explore the quantity and quality of the activities performed during software development. The analysis and categorization of the performed activities will further enable us to better link the data of the Git repository with the Jira repository in order to explore the evolution of the IIoT infrastructure's hardware and study the intertwinement of activities surrounding IIoT software and hardware.

Second, we will complement our computationally intensive analysis with qualitative inquiry in order to explore the mechanisms underlying the identified puzzles (Berente et al. 2019). Thereby, we aim to unpack key phases as well as punctuated changes across the development process. By interviewing stakeholders within the case organization regarding IIoT infrastructure, we can gain insights into critical decisions made during the IIoT infrastructure evolution. A synchronous analysis of the quantitative and qualitative data will enable us to explain IIoT infrastructure evolution. The qualitative data analysis will rely on an adapted grounded theory approach. Following the computationally intensive theory construction process suggested by Berente et al. (2019) and Miranda et al. (2022), we strive to address the presented puzzles and develop a theoretical model based on the uncovered patterns to explain the mechanisms underlying the evolution of digital infrastructures.

Acknowledgments

This research was funded by the Research Fund of the University of Liechtenstein (grant number F&E WI/75), and the European Union [EU Funding 2021-1-LI01-KA220-HED-000027575]: "Developing Process Mining Capabilities at the Enterprise Level".

References

Antinyan, V., Staron, M., and Sandberg, A. 2017. "Evaluating Code Complexity Triggers, Use of Complexity Measures and the Influence of Code Complexity on Maintenance Time," Empirical Software Engineering (22:6), pp. 3057–3087.

Basili, V. R. 1980. "Tutorial on Models and Metrics for Software Management and Engineering," in Proceedings of the IEEE Computer Society's Fourth International Computer Software & Applications Conference, pp. 278–289.

Berente, N., Seidel, S., and Safadi, H. 2019. "Data-Driven Computationally Intensive Theory Development," Information Systems Research (30:1), pp. 50–64.

Boehm, B., Abts, C., and Chulani, S. 2000. "Software Development Cost Estimation Approaches — A Survey," Annals of Software Engineering (10:1), pp. 177–205.

Bygstad, B., and Øvrelid, E. 2020. "Architectural Alignment of Process Innovation and Digital Infrastructure in a High-Tech Hospital," European Journal of Information Systems (29:3), pp. 220–237.

Fenton, N. E., and Bieman, J. 2014. Software Metrics: A Rigorous and Practical Approach, (Third Edition.), CRC Press.

Geraci, A., Katki, F., Lane, J., McMonegal, L., Meyer, B., Porteous, H., Radatz, J., Springsteel, F., Wilson, P., and Tee, M. 1991. "IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries," in IEEE Std, pp. 1–217.

Golder, S. A., and Macy, M. W. 2014. "Digital Footprints: Opportunities and Challenges for Online Social Research," Annual Review of Sociology (40:1), pp. 129–152.

Hanseth, O., and Lyytinen, K. 2010. "Design Theory for Dynamic Complexity in Information Infrastructures: The Case of Building Internet," Journal of Information Technology (25:1), pp. 1–19.

Hanseth, O., and Modol, J. R. 2021. "The Dynamics of Architecture-Governance Configurations: An Assemblage Theory Approach," Journal of the Association for Information Systems (22:1), pp. 130–155.

Henfridsson, O., and Bygstad, B. 2013. "The Generative Mechanisms of Digital Infrastructure Evolution," MIS Quarterly (37:3), pp. 907–931.

Koutsikouri, D., Lindgren, R., Henfridsson, O., and Rudmark, D. 2018. "Extending Digital Infrastructures: A Typology of Growth Tactics," Journal of the Association for Information Systems (19:10), pp. 1001–1019.

Lazer, D. M. J., Pentland, A., Watts, D. J., Aral, S., Athey, S., Contractor, N., Freelon, D., Gonzalez-Bailon, S., King, G., Margetts, H., Nelson, A., Salganik, M. J., Strohmaier, M., Vespignani, A., and Wagner, C. 2020. "Computational Social Science: Obstacles and Opportunities," Science (369:6507), pp. 1060–1062.

Lindberg, A. 2020. "Developing Theory through Integrating Human and Machine Pattern Recognition," Journal of the Association for Information Systems (21:1), pp. 90–116.

Lindberg, A., Berente, N., Gaskin, J., and Lyytinen, K. 2016. "Coordinating Interdependencies in Online Communities: A Study of an Open Source Software Project," Information Systems Research (27:4), pp. 751–772.

Miranda, S., Berente, N., Seidel, S., Safadi, H., and Burton-Jones, A. 2022. "Computationally Intensive Theory Construction: A Primer for Authors and Reviewers," MIS Quarterly (46:2), iii–xviii.

Modol, J. R., and Eaton, B. 2021. "Digital Infrastructure Evolution as Generative Entrenchment: The Formation of a Core–Periphery Structure:," Journal of Information Technology (36:4), pp. 342–364.

Montealegre, R., Iyengar, K., and Sweeney, J. 2019. "Understanding Ambidexterity: Managing Contradictory Tensions between Exploration and Exploitation in the Evolution of Digital Infrastructure," Journal of the Association for Information Systems (20:5), pp. 647–680.

Osmundsen, K., and Bygstad, B. 2022. "Making Sense of Continuous Development of Digital Infrastructures," Journal of Information Technology (37:2), pp. 144–164.

Østerlund, C., Crowston, K., and Jackson, C. 2020. "Building an Apparatus: Refractive, Reflective, and Diffractive Readings of Trace Data," Journal of the Association for Information Systems (21:1), pp. 1–22.

Pentland, B. T., Recker, J., Wolf, J. R., and Wyner, G. 2020. "Bringing Context inside Process Research with Digital Trace Data," Journal of the Association for Information Systems (21:5), pp. 1214–1236.

Sarker, S., Chatterjee, S., Xiao, X., and Elbanna, A. 2019. "The Sociotechnical Axis of Cohesion for the IS Discipline: Its Historical Legacy and Its Continued Relevance," MIS Quarterly (43:3), pp. 695–719.

Sisinni, E., Saifullah, A., Han, S., Jennehag, U., and Gidlund, M. 2018. "Industrial Internet of Things: Challenges, Opportunities, and Directions," IEEE Transactions on Industrial Informatics (14:11), pp. 4724–4734.

Tilson, D., Lyytinen, K., and Sørensen, C. 2010a. "Digital Infrastructures: The Missing IS Research Agenda," Information Systems Research (21:4), pp. 748–759.

Tilson, D., Lyytinen, K., and Sørensen, C. 2010b. "Desperately Seeking the Infrastructure in IS Research: Conceptualization of 'Digital Convergence' as Co-Evolution of Social and Technical Infrastructures," in Proceedings of the Annual Hawaii International Conference on System Sciences.