Platform Change: Theorizing the Evolution of Hybrid Product Platforms in Process Automation

Sandberg, Holmstrom, Lyytinen

Abstract

Recent research has taught us why platforms materialize and what their generic features and benefits are. However, research explaining how platforms evolve with new digital capabilities remains scarce. Most past research assumes that platforms emerge following a top-down strategy of modularization as to achieve flexibility for mass customization. In addition, it assumes "green-field development" – i.e. development characterized by how platform owners define and design new architectural rules for every platform, which delineate its modular structure and component interactions. In this paper we theorize through an exploratory case study the evolution of hybrid product platforms – those that include both physical and digital components and functionality and which need to maintain consistency across evolutionary steps during platform evolution. This is accomplished by tracing change in a process automation platform in a leading process automation company over a 40-year period. Process automation systems have been used since the early 70's for automating, controlling, coordinating and maintaining complex industrial processes including oil refinement, power generation or paper mills. They integrate original analog control systems and functions with new expansive digital functionality. We analyze the succession of the firm's automation system offerings as to detect the influence of different architectural principles that have guided platform strategies including the use of data abstraction or use of the Internet stack to organize the platform functions. We show that increased digitizing of the platform functions results in an increasingly loosely coupled architecture which promotes new stretch-fit patterns between increased technology variability and its alignment between use contexts which calls for constant changes in firm's strategy and economic logic. We illustrate novel challenges in managing hybrid platforms as platform providers need to cope with the dynamic growth of new digital and physical components. The study extends current platform theory by illustrating how digitizing analog components creates a new breed of hybrid platforms with specific platform design challenges. It also demonstrates that generativity – enabled by digitizing the functions of the control system – leads to increasingly distributed forms of control whereby the platform often evolves in unpredictable ways where new combinations of features are integrated due to the expansive scope and intensity of learning. Finally, it shows that platforms and strategy co-evolve through a stretch-fit pattern as digital capabilities increase technology variation calling for new strategic alignments with use and production contexts. These findings can be applied to a new breed of platforms emerging in such diverse fields as intelligent cities, power grids or new generation of traffic and power train systems.

Keywords: platform, platform evolution, architecture, digital capabilities, physical platform, digital platform, hybrid platform

1. Introduction

A platform, in general, is "a building block, which can be a product, a technology, or a service, that acts as a foundation upon which other firms can develop complementary products, technologies or services" (Gawer 2009: 3,4). Platforms are often viewed as multi-sided in that they mediate the world of technology with varying business logics. In this regard, they consist of building blocks that serve as a foundation for a firm to develop new complementary products, technologies or services (Gawer, 2009). Therefore a platform's architecture is expected to be evolvable in that it can adapt to unanticipated changes by re-using and re-arranging its components while allowing its peripheral components to vary. By doing so the platform allows increased functional variety and thus becomes evolvable – two characteristics sorely needed to compete during the age of customization and market uncertainty (Baldwin and Woodard 2009).

Recent research has extensively examined the conditions of the platform change and its effects on competitive strategy in a variety of industrial contexts, including the computing industry (Bresnahan and Greenstein 1999), the automobile industry (Sako 2009), and the aerospace industry (Brusoni and Prencipe 2009). Recently Yoo et al. (2012) also suggested that digitizing and its affordance of convergence is one of the primary drivers for platform change. They note "from one perspective, in order to harness the convergence and generativity made possible by pervasive digital technology, firms now innovate by creating platforms rather than single products." (Yoo et al. 2012, pp. 1400). The penetration of digital technologies into products and services and their success as witnessed by the history of Itunes or Amazon has recently heightened the role of digital platform strategies in many firms' innovation activities (Yoo et al 2010; Tiwana et al 2010; Tilson et al 2010; Yoo et al 2012). Despite the growing research on digital platform strategies, it remains unclear under what conditions digital platforms evolve and why they evolve. In particular, we know little about the effects of digitizing of previously physical platforms on their evolution – products built around digitizing previously physical aspects of products and processes. This process creates the need to relate a firm's physical platform to multiple new ecosystems. The firm now needs to source its products or services across multiple innovation domains such as devices, networks, contents, and services thus increasing its innovation complexity and diversity (cf. Yoo et al. 2010). Moreover, as Yoo et al. (2010) note, the inherent properties of digital technology challenge traditional organizing logics that underlie innovation and manufacturing of physical products which all assume hierarchically organized modular product structures. In consequence, physical platform providers need to now adopt and deploy new architectural principles that differ radically from those recognized in the past (see e.g. Henderson and Clark 1990; Baldwin and Clark 1997). In short these architectural principles assume a new hybrid architectural logic (cf. Yoo et al. 2010), because they cut across previously tightly integrated layers of innovation by separating: (a) devices from their control systems (Lee and Berente 2012) and (b) content from its application (Tilson et al. 2010). How these new loosely coupled principles influence, evolve and become embedded into hybrid product platforms and what their consequences are for the ensuing platform strategies have not, however, been addressed.

Platform evolution has been mainly explored in the context of market-based competition on double-sided markets (Rochet & Tirole, 2003, 2006; Parker & Van Alstyne, 2005), and related concerns for strategy management (Gawer, 2009; Gawer & Cusumano, 2002). Most existing studies are concerned with change in the scale of the platform and under what conditions the platform growth will 'ignite' (Evans 2009). In this article, we complement this understanding of platform change by analyzing qualitative changes in platforms rules and architecture and how they relate to strategy i.e. how the platform is positioned with regard to its use and production contexts. What is evident from the extant research is that platform evolution is associated with a co-evolutionary pattern between technology variation, use context variation and emergence of new strategic options to align these two i.e. whereby technology is made fit its surrounding context (e.g. Evans et al., 2006). A typical trigger for creating a pattern of co-evolution is ”technology stretch” enabled by digitization which, in turn, searches for a new context as well demands the strategy to comply (Barrett et al. 2012; Lee and Berente 2012). Recognizing such evolutionary gaps enabled by increased technology variation, we ask the following research question: How do hybrid platforms evolve as they become enabled by expansive digital capabilities?

By addressing this research question we pull together some building blocks for an integrative framework how platforms are likely to evolve as they become increasingly armed with expansive digital capabilities. We formulate the framework by conducting an exploratory longitudinal case study of the evolution of process automation system offerings in a leading process automation provider starting from the late 70's leading up to the recent announcement of a new kind of platform establishment. In particular we analyze how the automation control system gradually expanded its functionality as it embraced architectural principles that followed digital system design that we call loose coupling. While originally the process control system was designed to purely control physical production equipment and keep their behaviors within accepted physical tolerances. Due to its extensive physical embedding such systems have extensively long life cycles (over 30 years). Yet, its later evolution and new incarnations of platform features have added new and often unexpected digital functionality and radically changed its features, dynamics and ecosystem. The case study illustrates how the gradual emergence and evolution of a hybrid process control platform was punctuated by constant stretches between technology variation and new context variation and how this was enabled by constants shifts in platform strategy.

The remainder of the paper is organized as follows. Section 2 reviews related research on platforms, discuss physical and digital platform concepts and introduce the concept of hybrid platforms. We also outline current gaps in our understanding of platform evolution. Section 3 discusses the research design, data collection and analysis. Section 4 presents key findings of the evolution of the automation platform by tracing changes in platform features, depth of the digitizing architecture and changes in the technology 'stretch'. We conclude by noting the implications for our findings for platform evolution and strategies.

2. Related Research

In the following, we present related research on platforms and digital innovation, and review current frameworks for analyzing socio-technical co-evolution and growth patterns in strategy contexts.

2.1 Platforms

While the word "platform" has been used since the mid 16th century to describe both physical and abstract phenomena, it's meaning has since then expanded and changed, and the definition of "platform" in extant research is not necessarily unified (Baldwin & Clark 2009). We address this by conducting a systematic literature review of the platform literature in order to uncover the underlying theoretical logics. In so doing we begin with examining three streams of research.

The first stream is the "product platform" which is often examined from the viewpoint of a single focal firm in the context of physical product development. Meyer and Lehnerd (1997) provide a widely adopted definition of such a platform type: "a set of common components, modules, or parts from which a stream of derivative products can be efficiently created and launched" (p. 7). The fundamental logic underlying this stream of research is the idea of modularity, i.e. decomposition of a system into separable parts (Simon 1972). Research on product platforms is heavily founded on engineering of manufactured products such as consumer electronics. Empirical examples include how Sony introduced more than 250 models of its Walkman in the 1980s (Sanderson and Uzumeri 1995), Kodak's camera development during the 90's and the car industry (Robertson and Ulrich 1993).

The second stream is the "market platform" which extends the concept of the product platform into markets operating through the platform. This is aligned with the industrial economist view of platforms as mediators of transactions in two or multi-sided markets (Rochet and Tirole 2003). A key characteristic of this stream of research is that it emphasizes economic governance of the platform and value extraction. Governance forms here an integrated part of the actual platform, as exemplified by Eisenman et al.'s (2006, p. 5) definition: "A platform embodies an architecture – a design for products, services, and infrastructure facilitating network users' interactions – plus a set of rules; that is, the protocols, rights, and pricing terms that govern transactions". This stream of research has mainly focused on pricing mechanisms and how they influence such diverse markets as shopping malls, operating systems and gay bars.

The third stream is the "software ecosystem platform." A software platform is defined as "the extensible codebase of a software-based system that provides core functionality shared by the modules that interoperate with it and the interfaces through which they interoperate" (Tiwana et al., 2010, p. 676). The software ecosystem platform has the potential to attract new users, increase "stickiness (i.e. make it harder to change platform), accelerate innovation and provide sharing of development costs (Bosch 2009). Third-party developers often develop modules operating on these platforms and the ecosystem is susceptible to network effects. Loose coupling in the layered architecture of software ecosystems platforms allow module developers a relative freedom to innovate as compared to physical product platforms where functionality is inscribed by the platform owner (Yoo et al. 2010). Hence, architectural characteristics in digital technology implies that these platforms are susceptible to generativity, i.e., "a technology's overall capacity to produce unprompted change driven by large, varied, and uncoordinated audiences" (Zittrain 2006, p. 1980). Attracting both developers and users are key challenges for this type of platform, as a result research on software ecosystem platforms often has a sociotechnical focus and includes impact of architecture, governance and contextual dynamics (Tiwana et al. 2010).

Table 1. Types of Platforms

Type	Rationality	Control	Change	Research Focus	Exemplary papers
Product Platform	Modularity allows re-use and decreases complexity, standardization of platform combined with customization allows economies of scale and scope. The overarching goal is product efficiency and functionality.	Internal, architectural decisions ripple through the design process meaning that early choices remain strongly inscribed.	Re-design of the platform based on end-user functionality.	Engineering assumption of rational design process and clean slate design.	Simon 1972; Meyer and Lehnerd, 1997; Robertson and Ulrich, 1998
Market Platform	Re-use of infrastructure allows efficient transactions. Focus on market efficiency and transaction costs. Competitive advantages are achieved by attracting a large number of providers and customers through strategic decisions.	Control of access to market.	Re-design based on attractiveness for the two sides of the market	Economics and strategy. Pricing mechanisms Assumption of perfectly rational actors allows control through price.	Economides and Katsamakas, 2006; Eisenmann, Parker and Alstynem, 2008; Eisenmann, Parker and Van Alstyne, 2008;
Software Ecosyste m Platform	Shared functionality in codebase allows specialization, distribution of development costs and access to users. Commonality achieved through shared platform rather	Control through architectural rules (interfaces).	Re-design based on end-user or module-developers need of functionality, or changes in the technological base	Software engineering, information systems and innovation strategy. Assumption of co-evolution between architecture, contextual	Baldwin and Clark 2006; West and O'Mahony, 2008; Tiwana et al. 2010; Eaton et al. 2011

	than application area.			dynamics and strategy.	

A main focus of platform owners across platform types is to provide a set of core components and resources that facilitate the process of generating complementary assets (Robertson and Ulrich, 1998), which therefore increase the value of the platform. According to Baldwin and Woodard (2009), the reuse of such components and resources acts as a powerful economic logic. Economies of scale can be realized through increased production volume and the efficient use of complementary assets. Also, as is evident from the literature review, recent research on platforms has increasingly recognized the significant value of third-party developers and their contributions in platform innovation (Bosch 2009; Evans 2009; Remneland-Wikhamn et al. 2011). Their importance continues to be recognized as they build and sustain platform innovations with complementary assets in the form of software applications and services (Evans et al. 2006; Hanseth and Lyytinen 2010; Eaton et al. 2011).

Common in all three streams of platform research is a focus on the conditions under which platforms change and grow. Armstrong (2006) and Rochet and Tirole (2003) suggested that a key for technology platform change lies in that platforms enable multi-sided markets bringing together various types of participants (sides) such as buyers and sellers. Studies of such platform growth have used various lenses to understand platform competition (Rochet and Tirole 2003), emergence (Iyer et al. 2007), strategies (West 2003), strategic differences (Economides and Katsamakas 2006), and the role of complementary markets (Gawer and Henderson 2007). However, there is a paucity of research exploring change in types of platforms, linking the platform type to the generative mechanisms driving platform change. Understanding these dynamics are imperative in the case of platform evolution over a longer period, which by definition can involve multiple platform types and the generative mechanisms driving the transition from one platform type to another.

Advances in the functionality of information technologies and transformations in the way they are being fused with products and services has led some scholars to challenge conventional wisdom about the link between platform types to the generative mechanisms driving platform change. For instance, Tiwana et al., (2010) address the interplay of architectural design, platform governance, and the environmental dynamics of ecosystems on the evolution of platforms. In so doing they address key concerns in such interplay, such as the delicate balance of control by a platform owner and autonomy amongst developers. Evans (2009) observes that two-sided markets must secure enough customers on both sides – and in the right proportions – to give enough value to both sides and to attain sustainable growth. Drawing from cases such as Friendster and Facebook he discuss how they must achieve secure critical mass to ignite the growth of their platforms and how the failure to achieve critical mass quickly results in the implosion of the platform.

2.2 Physical and Digital Platform Architecture

In the academic literatures the term ‘platform’ has been applied to many things: products, systems, services, and technologies as diverse as durable goods (e.g.

airframes and jet engines, car underbodies and engines, printers, and power tool motors), operating systems, networking protocols, credit card systems, video game consoles, shopping malls, and dating sites (Baldwin and Woodward 2009). Most literature treats physical (tangible such as cars of shopping malls) and digital platforms (such as Ebay) in a similar manner. One reason for this is that the literature originates almost exclusively from fields such as new product development, strategy, and economics, where the world of bits is not conceived as being much different from the world of atoms (see e.g. Baldwin and Woodward 2009). No doubt, physical and digital platforms share many similarities in terms of their basic economic logic and valuation such as how fixed or variable costs relate or how network effects are realized. We remain doubtful, however, that theoretical predictions will hold the same for all critical topics on platform strategies for such a wide variety of phenomena. Especially, we are doubtful that the predictions are the same when we need to understand the change in platforms.

One reason for our doubt is that platforms differ fundamentally in their technological affordances, i.e. "what an individual or organization with a particular purpose can do with technology" (Majchrzak and Markus 2012). One of these differences is affordance complexity and variability and the consequent intricacy of the underlying platform (Tilson and Lyytinen 2013). Another is the pace and rate of variation afforded by the platform over time. Here physical and digital platforms differ radically and these differences can be largely attributed to the fundamental differences in the underlying material that makes up the technology. These fundamental differences are also manifested in structural patterns – architectural principles – that guide the design of physical or digital functionality on a platform, respectively. Physical platforms are typically designed using a modular architecture (Yoo et al 2010), while the architecture of digital platforms is in addition layered due to deep recursive organization of digital functional capabilities (Yoo et al 2010; Tilson and Lyytinen 2013).

Physical platforms can be composed of various material components where each component will carry out relatively a narrow range of specific functions and its alternative use value is close to zero (an engine is an engine). Therefore the components are harder to change and fit to alternative compositions and the conditions for creating such variability remain restricted. The main strategy to achieve such variability, as noted, is modularity which is about decomposing a platform (and its functions) into components that can be combined in multiple ways as to achieve desired functional variability (Simon 1996). To do so components need to be designed following the principle of relative interdependence i.e. high internal component cohesion and minimizing external connections. This is achieved through functional abstraction, information hiding and standardized interfaces (Baldwin and Clark 2000). This principle helps minimize negative effects of increased functional complexity associated with the desired increased variation.

This strategy offers also other benefits. Once the overall architecture has been determined design and manufacturing of identified families of components can be allocated to more specialized actors as to improve economies of scope. At the same time a growing variety of alternative combinatorial solutions helps achieve economies of scale and scope. Overall, through a modular architecture firms can carry out vertical disintegration (Baldwin 2008) as to improve scalability and flexibility. There are also drawbacks associated with the strategy, for instance there are additional initial costs for determining initial architecture. Decoupling of components based on functional hierarchy requires, however, that relations between components remain fixed and their relationships can vary only within a very strict set of limits. Therefore, the achieved level variability on a platform is always a matter of degree. This sets clear boundaries for variability and can create competency traps. This is one of the reasons why despite 25 years of extensive effort and research highly modular car manufacturing remains still a dream (Helper and Sako 2010). Due to these effects the range of variability and the rate of variation in physical platforms have remained somewhat limited.

In contrast, digital technologies offer unprecedented variability in terms of their scaling potential as witnessed e.g. by the exponential growth of social networking sites (Fisher et al 2013) and functional variability as exemplified by the fast paced growth of functions that have been made available on Internet (Hanseth and Lyytinen 2010; Tilson and Lyytinen 2013), or on specific sites such as Facebook (Fisher et al 2013). The reason for this increased variability is that digital technologies are basically highly abstracted computing capabilities (semiotic engines) that can be materialized on any medium capable of changing between a discrete state and signaling the current state. This abstraction is achieved by separating the physical unit that executes instructions- the processor- and the storage unit that holds instructions- the memory. The functional logic is then expressed in sets of instructions stored in the updatable memory and this set will always ultimately determine what the physical processor will do. The expressed functional logic of the physical processor can be reprogrammed at any time by changing the instruction set and thereby the functionality of the digital device can be altered easily. In short, the abstracted computing capability enables reprogrammable functions (Yoo et al. 2010; Yoo et al. 2012) and digital technologies can be described as general purpose technologies which have “the potential for pervasive use in a wide range of sectors in ways that drastically change their modes of operation” (Helpman 1998, p. 3).

Due to this abstraction digital technology can be changed simultaneously at multiple different layers: 1) through the installation of software that express new sets of instructions; and 2) through the change in physical capabilities of processors and storage units permitting faster execution and larger and faster storage. Moreover, due to unified and abstracted forms of expressing digital functionality (by innovations in software technologies such as compilers) the same set of instructions can be run across multiple different processors and storage technologies. In this way the functionality and variability of digital technology is never determined solely by hierarchical decomposition, but comes from a loose coupling (many to many mapping) between digitally expressed functionality and its physical materialization.

Another benefit of digital technologies that permits high variability is that they homogenize all data used in the computations into binary code form. This code can represent any type of data about the world, such as numbers, video, text or sound as long as the users (and their software) can agree what the data denotes (Tilson et al 2010; Yoo et al 2010). Thus compared to analog physical devices that are used to represent and store data, digital technologies are extremely adaptable in that they can be used for recording, storing and representing any information its users can acquire and represent about the world. As long as digital devices have access to instructions on how to decode the binary data, they can process it and offer new functions. As a result of such digitizing content can be easily separated from the material form – the medium – which increases the functional variation enormously.

Due to these fundamental material differences and possibility to integrate different types of data an architecture for digital platforms is represented as a layered system of separate functionalities, where each layer follows principles of modularity (Benkler 2006; Gao and Iyer 2006). This architecture is typically called a 'stack' consisting four loosely coupled layers as illustrated in Figure 1 (Benkler 2006): content, service, network, and device. The device and network layer can be further divided into logical layers and physical ones (separating physical execution of computing from its logical representation). Components in digital platforms become thereby loosely coupled (with many to many mappings) and can be re-combined into different functional sets using either interfaces between layers (Yoo et al 2010; Tilson and Lyytinen 2013) or gateways between different functions on the same layer (Hanseth and Lyytinen 2010). While the user of the digital platform often perceives each layer as a single 'functional' unit, it can be designed and used with a minimal level of consideration and knowledge of the other layers on the platform as long as information about the interfaces on the lower layer are known. This offers significant benefits in terms of economies of scale and scope and offers new sources of variability.

An additional outcome of separating functions into loosely coupled, interoperable layers is that this principle permits new functional combinations to emerge that were not foreseen at the design time. Indeed digital platform "exhibits a procrastinated binding of form and function (Zittrain 2006) meaning that new capabilities can be added after a product or tool has been designed and produced" (Yoo et al. 2012, p. 1399). Hence, while the core layers of the digital platform in itself remain stable (Tilson et al 2010), the loose coupling allows higher levels of variation due to generativity of digital platforms, i.e. an "overall capacity to produce unprompted change driven by large, varied and uncoordinated audiences" (Zittrain 2006, p. 1980).

Overall physical and digital platforms vary in terms of (1) boundaries and variation of use contexts (Yoo et al. 2010) (single purpose v.s. general purpose), (2) governance (Boudreau 2010; Tiwana et al. 2010) (centralized control v.s. open forms of control), and, (3) evolvability (change in the scale and nature of the platforms) (Kirschner and Gerhart 1998). Differences in couplings lead to differences in variation in the firmness of platform boundaries and use contexts. Digital platforms can serve a wider variety of purposes in various contexts, as exemplified by Microsoft Windows, whereas physical platforms such as car platforms can be flexibly designed in terms of how they variably meet some of the customers' needs, while they all fulfill the purpose of transport. Finally, components in digital platforms can be invented and then combined within and across layers in ways that were not foreseen in the original design time. Therefore digital platforms are often used in unanticipated ways and contexts.

The generative characteristic of digital platforms impacts their governance. While economies of scale and price competition among component suppliers affect attractiveness of physical platforms, the value of a digital platform is more often related to the growing heterogeneity in component suppliers (Boudreau 2010; Boudreau 2012). Further, decision rights and forms of ownership exhibit larger variety (Tiwana et al. 2010). Decision rights for both platforms are similar in the sense that both are enforced through control of interfaces; the difference is that digital platforms can adapt and use gateways or adapt to new reconfigurations even after the platform has been released. For example, Apple decided to open their handheld device for third party developers after “jail-breaked” versions showed enormous potential benefits. A similar adaptation is hard to achieve on a physical platform once interfaces are manufactured and sold.

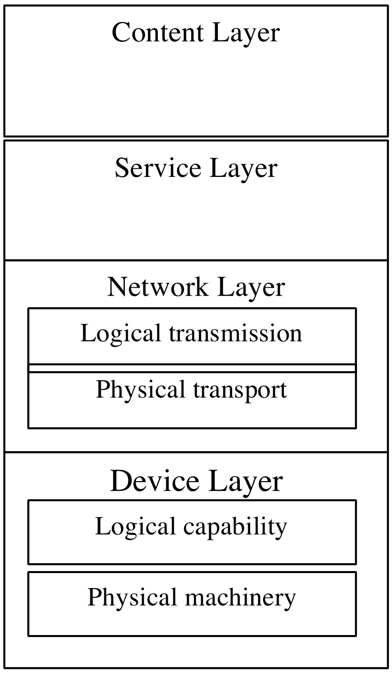
Finally, digital platforms appear to be more evolvable than physical platforms due to high level of functional abstraction. Once a physical platform has reached the end in its variation capability, it is phased out. Digital platforms instead are in a constant flux of change and adapted to new uses and contexts. As digital platforms are phased out, costs of adapting them are significantly lower and as a result parts of them can continue to function in environments and for purposes that are fundamentally different from what they were designed for.

Examples of both physical platforms and digital platforms abound in the literature. The former ones cover durable goods (e.g. airframes and jet engines, car underbodies and engines, printers, and power tool motors) and locales (shopping malls or gay bars); the latter ones operating systems, credit card systems, video game consoles, electronic auction sites, social networking sites or dating sites. Sometime the main functions served can be quite similar (gay bars vs. dating sites). When such pure ‘forms’ are present the varying principles that govern their use contexts, design, and governance are more unified and simpler. There is, however, as set of industrial products that integrate increasingly physical and digital components and related functionality. We call such platforms hybrid platforms in that their special characteristic is the intermingling of modular physical components with digital functionality as to increase the platform variability and evolvability. These platforms range from ‘intelligent’ materials (Internet of Things), to intelligent products, processes and environments. Examples include power transmission and exhaustion systems for engines (Lee and Berente 2010), process automation systems, intelligent cities (Desouza and Bhagwatwar 2012) and so on. In these platforms the states of the physical world are continually mapped to digital representations (with sensors embedded in physical components) and then the world is operated and controlled through a set of effectuators (digital to analog conversion).[1] These platforms are currently becoming more prevalent (just consider current interest in intelligent cities, transportation systems and power grids) and their unique feature is constant evolution and change. This combination calls for integrating design and governance principles from both types of platforms and thereby generates a set of tensions when the two architectures need to be combined. Therefore one of the key challenges for platform scholars is how do they evolve and what mechanisms influence their evolution. To understand these mechanisms we need to understand how technologies, contexts of their use and production and organizational strategies are related in ways that influence variation, selection and retention in a growing set of platforms functions, features and principles.

---

[1] In a limited sense this happens with all digital systems that interact with humans as human sense and operate only with analog representation. However, in these systems we analyze the boundary between analog and digital representation is part of the platform design i.e. what types of ‘states’ are mapped into the digital representation.

Figure 1 Layered digital architecture



2.3 Co-evolution of Technology, Context and Strategy: An Integrative Framework

In this section we review approaches that can be used to explain why and how hybrid platforms evolve. First we posit that the functions and variability potential of technology, its use and production context, and organizational strategy - the way in which features and variability are aligned to use and production contexts- are central in understanding why and how hybrid platforms evolve. Typically, technology has a close fit and therefore low variability due to its closeness with the existing practices (use and manufacturing contexts). As new technical opportunities emerge the elements are related in a new way to what Geels (2005) refers to as a stretch pattern. In this pattern technology variation increases so that it becomes misaligned with context and strategy. Overtime context and policy need to co-evolve as to select and retain technology variation. In a sense the original technology stretch pattern will be stretched for context and strategy. Indeed, the platform can at any point of time provide a pool of common resources waiting to be expanded and varied. Thereby the platform owner can generate additional value by varying platform use and production contexts. In this regard platform strategy is about building stable ecosystems where various actors with differential knowledge, assets and interest collaborate and compete as to change either the scale or the nature of platform and to increase its environmental 'footprint' or fit across use and production contexts (Evans et al. 2006; Gawer and Cusumano 2008).

The relationships between technology, context and strategy – especially when digital technology potential is being leveraged – are highly dynamic and poorly understood. In particular, we know little how technical, contextual, and strategy elements combine for successful evolutionary outcomes with hybrid platforms as the platform owner needs to apply contradictory and varying architectural rules and mechanisms for variation, selection and retention. Accordingly, there is a need for research how a hybrid platform's architecture and related strategic choices influence its evolutionary trajectory (Tiwana et al. 2010; Yoo et al. 2010).

To understand this puzzle we draw on evolutionary models that seek to detect and explain the emergence, development, and demise of organizational forms, routines or strategies - such as platform development. Indeed, evolutionary economics has for some time stressed the Schumpeterian nature of processes surrounding complex phenomena like hybrid platforms and the essential role of variability in assets and knowledge to understand underlying evolutionary mechanics (e.g., Nelson and Winter, 2002). Platform research therefore has to appreciate the subtle two-way interactions between technical, contextual, and strategy elements as it explores how and why platforms evolve. The core of the co-evolutionary argument is that the current nature of technology platform (and its variability) sets boundaries to the patterns of innovation (variability) and that the subsequent learning processes for production and use will depend on prior levels of technological variation. The trick here is that adoption of hybrid forms will increase the variability in the order of magnitude when compared to pure physical platforms. Innovations will be more varied and will evolve through multi-level learning processes that vary the levels of variation, selection and retention associated with platform. The platform owner's task, accordingly, is to exploit any existing pool of platform components and capabilities, and second to harness new digital resources and capabilities to as increase variability and selection. This will happen by constantly stretching both digital technologies (dramatic variation of features and allowing radically new combinations) and expanding its use and manufacturing contexts. This will ultimately influence how well platform strategies will align with technology and context.

We will pursue these issues next in our study of hybrid platform evolution. In particular, we will address the need for fit between platform and its environment while paying a particular attention to the learning processes involved in reaching a functional fit. More specifically we will explore how the co-evolution between technology, context and policy follows a fit–stretch pattern.

3. Research Site and Methods

To understand the dynamic growth pattern of the platform and it's eco-system, our inductive case study involved data collection at ABB, third-party developers and customers in the mining and pulp industries. Data was collected through interviews, site and system demonstrations, design workshops and document studies in three phases.

3.1 Research Site

In order to examine platform evolution we chose to study distributed control systems that are used in complex industrial processes. These systems have been in use long enough to illustrate tensions between an installed base and expansive digital capabilities. They have also undergone radical change due to the increased digitization of production control over the last three decades. Distributed control systems are used to manage complex processes automation within industries such as mining, pulp, oil and nuclear energy. These use contexts are characterized by the use of heavy machinery and "extreme production environments" – in terms of intense noise, heat, dirt, vibrations etc – which means that high reliability in platform performance is critical. While industrial processes are managed through expansive digital capabilities, the actual processes remain very much dependent on physical equipment. We conducted a longitudinal study of the evolution of process control system offerings in one of the largest suppliers, ABB, starting from the 70's leading up to the recent platform establishment. In particular we analyze how the control system design gradually expanded in functionality and how it simultaneously embraced architectural principles that followed digital system design that we call loose coupling. While the process control system originally was designed to control physical production equipment with long life cycles, its later evolution and incarnations have added totally new digital functionality to the platform and radically changed its ecosystem.

Distributed control systems are built up of interconnected digital processing units that steers a dynamic manufacturing process (as compared to a discrete one) based on events. These systems can be represented in layers and they are distributed in the sense that that the controller elements are not centralized, instead they are located in the production environment and steers a restricted part of the process. Networks for communication connect these controller elements for monitoring and coordination purposes. Data is integrated in the distributed control systems that both controls production according to programmed logic and offers monitoring services for operators. At the lowest level of the system, the process level, sensors and actuators record signals and transmit them to controllers in real time. Based on these signals, controllers respond according to the preprogrammed logic and signals actuators to execute commands that change the process. All of this happens in milliseconds, as system must respond to process changes swiftly. Modern control systems also communicate with enterprise applications on issues such as inventory, produced volumes, maintenance orders etc. To provide such integration without compromising security systems are often designed with a demilitarized zone (DMZ) where data and services can be exchanged.

In this paper we describe and analyze platform evolution through the Purdue Reference Model as applied in the Instrument Society of America's (ISA) standard 99 (Kjaer 2003; CISCO 2013). The Purdue Reference Model (PRM) describes functionality and is widely applied as basis for segmentation of hierarchies in descriptions of manufacturing system. The ISA standard 99 adapts these hierarchies to industrial automation and control systems(CISCO 2013) as illustrated in Table 1. Industrial automation systems have four main duties, out of which all levels in the PRM are involved in assuring plant coordination and operational data reporting, and, system reliability and availability assurance. The responsibility for control enforcement is instead distributed to levels 1 and 2, and production scheduling to level 3, 4 and 5. To be able to perform these duties system architecture often include a demilitarized zone where data and services can be exchanged while lack of direct connections heightens security.

Table 1. Functional Model of Process Production Systems: Purdue Reference Model as Applied in ISA-99 (adopted from CISCO 2013)

Zone	Layer	Description
Enterprise Zone	Level 5: Enterprise Network	Centralized IT systems and functions, Enterprise Resource Management, business-to-business and business-to-customer services
	Level 4: Site business planning	Extension of the enterprise network, basic business administration performed through standard IT services. Access to Internet, e-mail and enterprise applications such as SAP. Non-critical plant systems such as manufacturing execution systems and plant reporting such as inventory.
Demilitarized Zone: Provides a buffer zone where services and data can be shared between Manufacturing and Enterprise zones		
Manufacturing Zone	Level 3: Site Manufacturing Operations and	The highest level of the distributed control system, manages plant wide automation functions. Reporting such as cycle times and predictive maintenance, detailed production scheduling,

	Control	asset and material management, control room workstations, patch launch server, file server. Domain services such as Active Directory, DHCP, DNS, WINS, NTP. Staging area for changes in the manufacturing zone and share data and applications though the DMZ. These applications are primarily based on standard computing equipment and operating systems. Tend to be aligned with standard IT technologies, hence implemented and supported by personnel with IT skill set.
	Level 2: Area Supervisory Control	Applications and functions associated with supervision and operation of each area such s operator interfaces, alarms, control room workstation. Communicates with controller in level 1 and share data with level 3 and/or level 4 and 5 through the DMZ
	Level 1: Basic Process Control	Controllers that steer automation of process based on input from level 0.
	Level 0: Process	Input and output units such as sensors, actuators that measures and perform the functions of the manufacturing system. Control engineers typically design and implement solutions that are typically remains unaltered for a longer period.

3.2 Research Methods

Our longitudinal case study focuses on the evolution of the platform over time, from the infancy of ABB's basic control system in the 70's up to today's eco-system that encompasses digital capability across all six layers of process manufacturing. As illustrated in figure 2, our research project consisted of four phases from 2007 to 2013. We collected data in three of these phases (2007, 2009 and 2011-2012) through formal interviews, informal discussions, observations at production sites, internally and publicly available documents, and, participation in workshops on how to design the platform's eco-system. We analyzed the data in four rounds of coding that progressed from understanding the research site and situated practices, to specifically targeting platform evolution and expansion of digital capabilities.

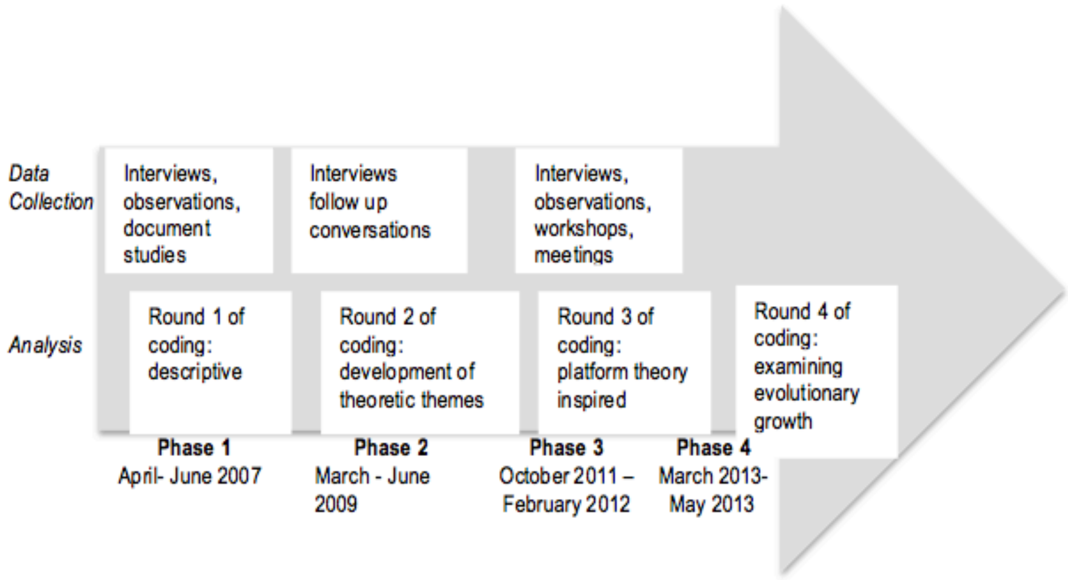


Figure 2. Research Phases

Our goal in the first phase was to examine an implementation project of an information system (IS) that should provide data on the installed base at customer's sites. Specifically, we were interested in maintenance of digital products, design challenges in interorganizational systems, and, potential effects on product and service sales. The first author spent five days at five different sites, three ABB offices and two mining production plants. Seven formal interviews were conducted out of which five informants were ABB employees in roles such as sales manager, unit manager mining automation, project managers and process engineering, and three with informants from a large mining company in in professional roles such as site IT manager and site maintenance manager. The interviews were recorded and transcribed and lasted on average an hour. Our interview questions focused on understanding architectural changes in systems, related adaptations of business logics and information processing regarding installed base of digital components. In addition to formal interviews, non-work time such as lunches, dinner and transportation time was spent with informants between interviews and plant tours. After the interviews, follow up questions were discussed with informants to further clarify remaining issues. ABB also provided access to extensive project documentation and system demonstrations. A first round of coding took place during the data collection as we developed descriptive categories of change in automation systems.

In the second phase, we conducted another round of coding to develop our understanding through theoretically grounded themes such as ”boundary spanning practices” and “system development innovation”. We also conducted three more interviews and collected further documentation of the IS project, which by this time had been halted. In autumn 2011 a project on development of the platform's eco-system was launched, the basic idea was to examine how to increase amount of third-party developers solutions available and certified in the eco-system. This initiative was the launching point of this research project´s third phase. We performed formal interviews (10), participated in meetings and workshops and collected documents from ABB, three process industry companies (two in mining and one in pulp) and third-party developers. During this phase we also coded the data according to platform theory inspired themes. Specifically we looked for comparison of context, strategy and technology across time in the platform's history and developed tables for comparison (Eisenhardt 1989).

The fourth phase consisted of data analysis in which we specifically addressed platform evolution. We re-examined and coded the data according stretch-fit patterns and punctuations of equilibriums (Gersick 1991), i.e. periods of more change as compared to periods of relative stability where only incremental changes are introduces. In our analysis, these punctuations represent the shift to a new era. We paid close attention to what drove these punctuations and resulting stretch and adaptations. What emerged was a specific understanding of the ways in which technology, strategy and context evolved throughout the hybridization. Table 2 provides an overview of the data in this study.

Table 2. Data Summary

Source	Interviews	Site Visits and Observation	Workshops	Documents
ABB	10	Four units including R&D center	Participated in three workshops on ecosystem development	Internal project documents and publicly available reports, product pamphlets and presentations.
Process Industries	7	Two mining sites and one pulp	Participated in three workshops on ecosystem development	Site descriptions and summaries of information processing and applications in plant
Third-party Developer	4	N/A	Participated in three workshops on ecosystem development	Documentation of solutions and companies
Total	20			

4. The Evolution of a Platform

Process automation was up until the eighties controlled through analog systems that largely was custom built for a single plant. Two important architectural shifts took place during the eighties: suppliers increased re-use through modularized systems and the process layer was digitized. This was the first steps towards todays hybrid platforms for process automation that are highly complex and integrated systems, heavily dependent on digital technology to react to changes on a millisecond basis and represent status of processes in real time. In this section we trace the evolution of ABB's process automation technology, as illustrated in figure 3, leading up to the release and implementation of the integrative hybrid platform.

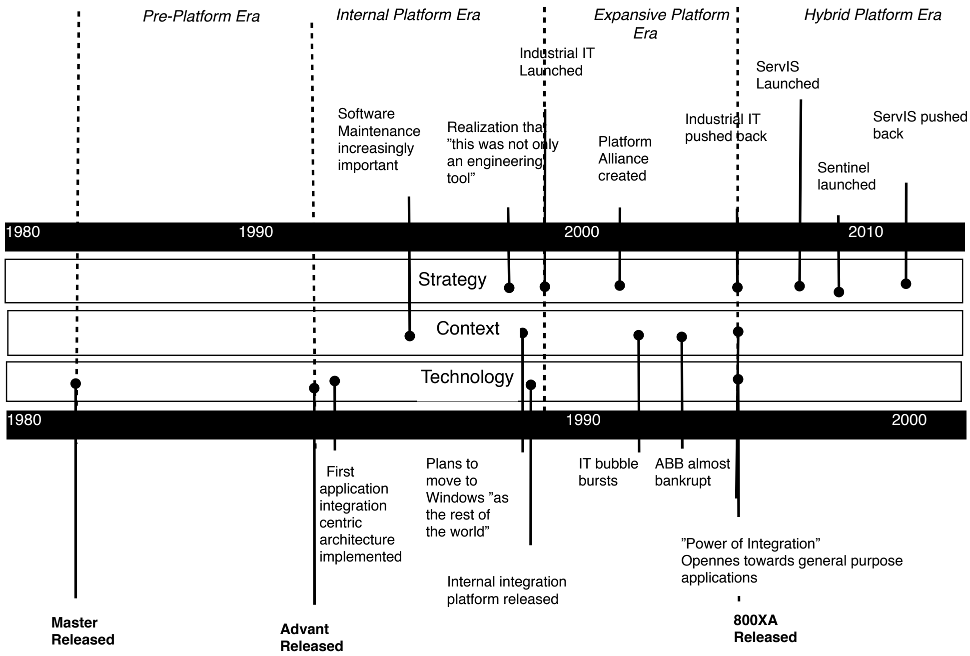


Figure 3. Platform Evolution

4.1 Pre-Platform Era: Re-Use Through Modularization

ABB is one of the leading suppliers of process automation equipment and has been in the market since the infancy of process automation systems in the seventies. These early control systems were relay based systems and were steered by manually switching currents, e.g. pushing buttons at large instrument panels that cut or activated electrical currents to production equipment. Information about the current status of the systems and processes was provided through light bulbs at the large instrument panel (which could cover a large wall). The relay based control systems had some serious limitations, they required a lot of space (the size varied according to required functionality), changes in the systems required physical re-wiring of cables and the possibilities to revise functionality was highly dependent on available space. The documentation of all the wiring was expensive and often insufficient, leading to large maintenance costs. Further, they did not have any substantial communication capabilities; instead they were limited to signaling alarms through lamps at the instrumentation panel. An increased demand on functionality towards the late seventies and early eighties had also increased complexity in the systems, leading to higher costs for both adapting and maintaining them.

The Master product family, which was released in 1983, was ABB's first step towards a unified platform for process automation. The Master system was centered on digital PLC systems and microcomputers that started to replace relay based control of the process in the late seventies and early eighties with digital capabilities.

"Master was the first system, we still have them, it was basically a PLC. It was here that the digital technology was introduced, we replaced relay based technology."
Process automation manager Pulp mill

Digitizing the control layer de-coupled control of physical machinery from the actual wiring, as a result designing the logic for process automation systems was done by adapting the software. The rationality behind this digitization was increased standardization and re-use of physical modules and interfaces.

"The ABB electronic and computer system, Master, is built up of catalogue-listed modules of hardware and software providing great flexibility and facilitating control system design. The application functions are solved in the software, which means that the hardware is limited to a small number of standardized and thoroughly tested p.c. boards. This also implies that the number of spare parts that need to be stocked is kept to a minimum."

"Relay-based control equipment is composed of several components. A great deal of the design work involves the interfacing of these components to each other. The larger and more complex the control equipment is, the more interfacing work there will be. A great deal of cabling and a great many connections are required to transmit all the requisite information in relay-based control equipment. By using Master, the interfacing problems are reduced to a minimum."
Master product pamphlet

The Master product family digitized steering of physical production equipment and provided significant advantages by reducing technological complexity, hence simplifying maintenance, adaptations and capacity extension. By replacing physical hard wiring of logics with programmable software, hardware design was increasingly standardized. Standardization of components and modularization, enabled by replacing analog capabilities with digital ones, allowed both ABB and customers to streamline current operations without seriously affecting current social structures and organization. Large portions of the system were custom built by ABB, for example they developed the operating system, database, hardware and software for the graphical interface. Such in-house production allowed ABB to take a user oriented approach to product design. Architectural changes were aimed at using existing physical production equipment more effectively. The result was a robust architecture:

"The first product was Master in the eighties which streamlined operations but did not change things dramatically. These products were developed in-house, we made everything, we built the operating system, the database, hardware and software for graphical interfaces, everything. It meant that we had pretty good control over these products. This is a true story, when we ask some of our customers about what the biggest problem with this system is they answer that Master and the Advant controller never malfunction, so they haven´t got a clue what to do when they break. They worked, extremely reliable."

4.2 Internal Platform Era: Developing Digital Capability

As ABB and customers started to appreciate potential benefits from sharing information throughout processes and organizations, openness became a factor. In order to leverage substantially improved computing capabilities in the IT industry, and to increase re-use of components in layer two and three, ABB released the object oriented Advant platform in 1992. While the Master product family allowed re-use of engineering solutions for physical coupling, and digitized level 1 basic process control, all commissioning of automation systems in terms of representations and logical design was done from scratch. When adding layers, and hence expanding the platform upwards in the process automation hierarchy, ABB faced important decisions in terms of what sourcing strategies, i.e. what to produce in-house and what to procure from suppliers.

While ABB developed most of the components in-house for the Master product family, it became increasingly clear this was not a sustainable strategy. The IT industry was developing rapidly in the late eighties and early nineties both in terms of business models and technological performance and in order to remain competitive, ABB needed to tap in to this innovation. ABB's solution was to use Unix as operating system and hardware from Hewlett Packard in their workstations. More importantly, ABB decided to increase the use of open standards widely applied in the IT industry and develop an object-oriented architecture that integrated data into a common database. Compared to the Master product family, which mainly standardized the lower levels of process automation systems, ABB now sought to re-use both externally developed technology and in-house developed engineering solutions such as libraries. Consequentially, the technology now became subject to development in the IT industry for off the shelf solutions.

Advant became a huge success; from 1992 to 1994 ABB doubled their market share. Integration was a key concept in developing Advant and it was implemented through standardization of system design, and physical and logical interfaces. The technological structure was designed with a set of stable modules necessary for all installations that could be integrated with a large number of add-on modules, i.e. a platform based architecture. In terms of functionality, the platform promised increased integration both for physical interfaces but also in regards to information sharing across boundaries at the production plant:

Functional integration
the same products and concepts are used in a uniform manner for all kinds of control, including logic, regulatory and supervisory control.
Structural integration
all products, from small controllers close to the process to advanced operator stations for central control, are parts of a system which is expandable step by step, from small scale to plant-wide automation.
Informational integration
the automation system is seen as an integrated part of the total computing environment at a plant. Thus information can easily be shared and distributed between all parts of an operation.
Advant "Product Guide"

As control systems became increasingly interconnected, system hierarchy and distribution became important. In order to ensure that the system could respond quickly enough, and that the system structure was kept clean, functionality was distributed to local controllers, workplaces and batch stations. This loose coupling between components provided flexibility in customizing, maintaining and expanding systems. Furthermore, loose coupling allowed redundant components in controllers and networks, and hence improved possibilities to design for reliability. While improvements in the process control layer was restricted due to physical limitations in the actual production equipment, the operator interface was radically improved as the Advant platform replaced basic text-based interaction

"With the Advant system and the AC450 controllers you could create object pictures. The graphical capabilities were improved towards the operator but in terms of process automation, well there´s not that much besides controlling the process you can do so you focus on other types of functionality, towards operators, towards MES [Manufacturing Execution System] systems and so on. System reliability was also improved, e.g. redundant bus technology was introduced in the plant during this era"
Process automation manager, Pulp mill

The most important functional features implemented in the Advant platform were arguably aimed at simplifying engineering. An object-oriented architecture allowed re-use of data and through modules with function blocks and process graphics the need for programming logics was reduced. The system also provided self-configuration by recognizing stations that were added without any configurations from engineers. Symbolic references and automatic adoption of new units in combination with isolation of application programs from communication technology would allow extension and re-structuration of the system through configuring modules, as compared to hard coding the configuration. Based on these functions, the Advant platform introduced re-use of digital resources in the engineering of production automation:

"The Aspect Object concept is the common base for the set of tools providing efficient engineering methods to support process automation projects. Each project can be broken down into smaller and smaller components called Aspect Objects. At the low level of the project structure these Aspect Objects can be individual signals, connection terminals, motors, etc. Aspect Objects are filled with information that describes different aspects of the corresponding real world object. These Aspects range from CAD drawings and functional descriptions to control programs. For easy access to the information there are different Views on the Aspects. The ability to include dynamic data in the Aspects gives the concept a very high flexibility. […] Import of existing data into the engineering environment is well covered, allowing coexistence with existing tools and methods. To achieve a high engineering efficiency and quality one time data entry is supported."
(Advant Open Control System, Product Guide)

4.3 Expansive Platform Era: Generative Automation IT

In the years following the release of the Advant platform, ABB initiated a learning process in terms of both effects of technology and strategy. ABB gradually increased the platform's functionality through upgrades and by releasing related modules. During the late nineties and beginning of the millennium the digitalization came to affect ABB's overall strategy in unforeseen ways. As the platform gained substantial digital capabilities, product boundaries were no longer defined by physical restrictions. Instead, the blurred technological boundaries opened possibilities for new use contexts and strategic positions.

While the Advant platform was an immediate sales success for ABB due to greatly improved functionality, it soon became apparent that the architecture had some significant flaws.

"The Advant platform was launched in 1992 and received a very warm welcome. After a couple of years we wanted to upgrade it and release a new version, that was when we ran into trouble. The system integrated information, data, but we could not get other actors to change their data format which meant that we had to assume responsibility for data storage for a lot of external systems. When we wanted to continue to develop the systems we had to take a larger and larger responsibility for developing others stuff. We realized pretty soon that it wasn´t sustainable"
R&D Manager, ABB

ABB R&D had to reconsider the architecture, the engineering group agreed that they needed to find a way to sustain the benefits from increased integration while avoiding responsibility for adapting to all changes application developers did in their systems. Pretty soon they agreed that the solution was to integrate applications, not information. The idea was to provide an adaptation module that could be changed according to changes in application APIs'. These ideas were developed already during a two-day workshop in involving R&D personnel from different countries and implemented in an engineering tool 1994. During this period thoughts applying this integration strategy in other use contexts were born.

"About this time we started thinking about how this idea of holding information together was interesting for a lot of actors, not only engineering. It´s interesting for operators, maintenance, production managers etc. so we changed the name into Automation Object, and then we started to think about the next generation of control systems."
R&D Manager, ABB

The idea that the integration architecture had larger application areas than only engineering, or for that part automation, gradually emerged among the R&D group. In the late nineties the group felt that this architecture had endless potential, "this is not only about ABB" and in 1999 the name was changed once again into Aspect Object. Over these years, the use context that the development team had in mind gradually grew:

"The basic architectural philosophy emerged over these years, at first we thought we were developing an integration platform for engineering tools, then for connecting all the different parts of the control system, then to integrate different products and units in ABB, and eventually to integrate the whole world"
R&D Manager ABB

In the late nineties ABB released Aspect Integrator Platform (AIP) built to integrate all functionality in the control system. ABB also released software development tools and started to build integrations towards general-purpose applications such as MS Office. The software development tools were later used to develop the third control system generation. As within so many other companies the enthusiasm was high in ABB for the possibilities of IT and the talk was that "we are really an IT company"

"There was a feeling of euphoria about this concept in ABB during this time, I´m not sure it was a positive thing, I mean this was something that we had dreamt up during a workshop that grew and grew and grew. I remember an annual report from this time, ABB had about 200 000 employees at that time, the only product mentioned in the whole annual report was AIP"
R&D manager ABB

Based on the powerful technology, ABB released the new brand name Automation IT. The strategic intent was to leverage the integrative capacity and provide certified solutions for a wide range of use contexts that customers could trust were compatible. These solutions should be both the ones developed in-house and by external suppliers. In spring 2000 ABB top management had an important meeting where the manager for the automation division declared that "*Automation IT is the future. Within a year, all products from my division must be Automation IT certified*". Enthusiasm among participants was high but according to one of the R&D managers developing the Automation IT concept nobody really knew what this certification meant. The automation unit manager contacted the control system R&D group and told them about the certification concept and asked them develop an appropriate strategy. The result was a four level integration certification, the higher the level the tighter the integration. Level zero was developed since the manager for the power division wanted to be a part of Automation IT, the problem being that they had no software. Hence level 0 came to mean that all information about the product is available so that it could be integrated. That is, all information such as service manuals, maintenance information, user manuals, calibration information etc.

Through the Automation IT concept ABB started to explore platform logics also in terms of strategy, i.e. as technology separated the platform from the pre-specified use context, traditional business positions were no longer given. ABB saw potential to make profit from licensing agreements and/or certification of products based on the Automation IT concept. Strategic discussion were also based on the fact that the automation industry was not the only ones needing integration, the construction and surveillance industry were seen as two possible targets and talks were initiated with possible partners. ABB realized that competing in new use contexts was a game changer and formed an alliance with partners such as Microsoft, Intel and Accenture to realize the potential benefits of an integration platform across industries. However, in 2002 all intents to expand came to a halt as ABB almost went bankrupt, as a result all but core operations were cut. While Automation IT remains a strategy and brand name within ABB, it´s role has been greatly downplayed.

4.4 The Hybrid Platform: Integrating Technology, Context and Strategy

Although the strategic plans to move the platform into new industries were largely abandoned, ABB kept developing the technology. Late 2003 they released the third generation control system Integrated Process Automation (IPA), built on the architectural vision from AIP. While the capabilities of the Advant platform emerged through a gradual learning process on the challenges and opportunities enabled by loose coupling and integration, IPA was designed for plant integration. IPA further extended the integration capabilities and scalability of process automation systems by leveraging available standards and off-the-shelf solutions in the IT industry. IPA increased capacity in a number of computational areas but the most important features were that layer 3 became a focal point of the process control and that functionality in layer 4 was improved. Technological advances in the general purpose IT industry also enabled richer data representations such as video surveillance of processes, off-site control and maintenance and virtualization of servers. ABB had by this time reached a substantial installed base with digital capabilities that began to be quite old, by building the platform on general purpose IT they also had less control than ever before over evolution of the platform. Being in a high-reliability industry with extremely long investment cycles for physical production equipment and witnessing an increased price competition over commoditized physical products, ABB faced new challenges in terms of software maintenance, managing a growing installed base, and developing related business models.

The arrangement with increased reliance on third-party suppliers of core modules of the control systems has led to fundamental changes. While the expected lifespan of logic in the first generations of control system was typically 10-15 of years, IPA hardware and software has a pronouncedly shorter lifespan and requires continuous maintenance. Due to the increased use of third-party products and open standards, and a high level of integration, they are increasingly affected by changes in the ambient IT environment. This rapid and continual innovation results in a steady stream of new technological solutions with better performance, making the existing ones obsolete and incompatible:

"When shifting from making all products and systems ourselves to using Windows, both ABB and our customers have ended up in a situation where we do not control the development of IT products. We are purchasing hardware from third-party suppliers and their products have their own life-cycles. […] Furthermore, updates and patches are frequent for Windows systems in home PCs, and the situation is the same with our systems."
Project manager, ABB

This dynamic presented ABB with new challenges regarding maintenance, the supply of spare parts and ensuring that software is compatibly updated. An example is how the local IT manager at PulpCo was unable to find the spare parts he needed, the only specimen available was owned by a technological museum. While digitalization provide new functionality for process automation, risk avoidance is the main strategy employed by ABB's customers due to extreme costs associated with system failure:

"You don´t want to install updates, to do that is a disturbance, there are always risks involved with installing updates. We minimize updates and only make changes in the software when there are obvious problems or when the version of Windows is so old that it is not supported anymore. That is the only circumstances we do it, I cannot recall that we have updated because we need new functionality."
IT Manager MiningCo

As the platform has gained digital capabilities, installed base management is increasingly challenging. Third-party sourcing and commoditization of IT reinforces ABB's role as guarantor of quality, and, compatibleness of software and hardware modules. To perform this function, it is imperative that ABB has information on versions of hardware and software currently in use at production sites. After the millennium, ABB has carried out a number of projects aimed at increasing their knowledge on installed base at customer sites in order to increase reliability and sales of services and products.

"We did an inventory in 2000 when we went through all the customers systems and created our own database, which was great for a year or so. Then the customers start replacing parts, or it can be some consultant doing it or some other company. Then the information we created is basically useless. This kind of information is great but updating procedures need to be more or less fully automatic. Every little thing has some kind of intelligence, everything with a plug has firmware."
Project manager ABB

So far they have not succeeded, customers systems are dynamic and since much of the products in use are old, they have limited communication capacity. Hence, ABB cannot implement automatic updates without customers replacing substantial parts of old production equipment, a strategy that is not feasible due to the large costs involved.

Based on increased competition and lower margins on IT products, services are an increasingly important income for ABB. Such service delivery is often related to information on both installed base and ABB customized and certified solutions. An example is how they sell software subscriptions to customers ensuring access to the latest certified solutions and information on compatibility problems. ABB also sells regular maintenance of IT systems such as twice a year going through the state of all hard drives in the production site. They are also trying to increase the amount of service sold in terms of engine maintenance, pre-paid consultant hours and ensuring a satisfying level of spare parts available in site stock. In all these activities, IPA is the fundamental block that ABB positions themselves to create an attractive eco-system around for process automation. E.g. while IPA itself might stand for 5% of total revenue from projects, the brand on the control system often decides which company that gets to deliver other products and services to the customer. Another example is how process industries in Sweden has created a joint group for discussing ABB automation issues and affect their R&D, the one thing that decides whether or not the process industry is seen as an "ABB customer" is whether or not they use their control system. With IPA, ABB has taken the final steps towards implementing a digital architecture in terms of controlling processes. Nevertheless, these processes remains anchored in the physical world and performed by expensive, high-reliability equipment. This hybrid logic implies that ABB needs to manage both the logics of a physical, modular platform, and a digital layered one. So far ABB has been able to adapt to the co-evolution of the platform's of technology, context and strategy. As it seems probable that IT will keep evolving faster than much of the physical production equipment, it remains to be seen how they can develop appropriate strategies for managing such stretch.

Table 3. Platform Evolution

Era	Architectural Principles	Technological computing components	Platform Logic	Platform Application Area	Value Proposition
Pre Platform 1983-1992 (Digitization era)	Re-use through modularity	Data chips on devices PLC	Product Logic Full control	Physical	Product units value
Internal Platform 1992-1998	API based on objects/ aspects Connecting people	Unix, analog/digital converters, network	Reuse Economies of scale and scope, openness, variability Control of system	Physical/digital mapping to abstraction	Integrator capability Automation of maintenance control Cost/reduction/ reliability
Expansive Platform 1998-2003	Unified powerful integration platform for Automation IT	Windows, unified powerful API:s, Internet	Value added services System platform for running application Control *of connectivity*	Digital integration	Effectiveness of automation systems, organization. wide integration Cost? Variability
Hybrid platform 2003-	Value integrated platforms for special vertical sectors	SOA, Apps integration, provisioning	Market for application, integrators, insert developers *Control of Access*	Digital expansion approach	Rents Value added service

5. Discussion

Extensive research on platforms provides a thorough understanding on why they materialize, what their features are and how they provide value. However, we know less on how specific architectural principles, and in particular digital ones, affect their evolutionary trajectory.

We conducted an exploratory longitudinal case study of the evolution of process automation system offerings in a leading process automation provider starting from the late 70's leading up to the recent announcement of a new kind of platform establishment. In so doing, we explored in detail how the co-evolution between technology, context and policy followed a fit–stretch pattern and we carefully examined the learning processes involved in reaching a functional fit between platform and its environment. Studying how the automation platform evolved as it became increasingly armed with expansive digital capabilities allowed us to analyze how the relationships between technology, context and strategy co-evolved. These issues are highly dynamic and poorly understood, and we know little about how technical, contextual, and strategy elements combine for successful evolutionary outcomes with hybrid platforms as the platform owner needs to apply contradictory and varying architectural rules and mechanisms for variation, selection and retention. Against this backdrop, we focused on how hybrid platforms evolve as they become enabled by expansive digital capabilities. We contribute to prior literature by highlighting three important characteristics in hybrid platform's logic: strongly evolutionary characteristic, dynamic growth, and, tensions between generativity and installed physical base.

First, our study extends current platform theory by illustrating how digitizing analog components created a new breed of hybrid platforms with strongly evolutionary characteristic and specific platform design challenges. These hybrid platforms come with specific tensions since they are subject to both generativity and restrictions from the installed physical base. Accordingly, managing them requires understanding of digital and physical product logic. In our study, both digital and physical characteristics of process automation systems also created a strongly evolutionary characteristic of the platform. Digital characteristics promoted variability through multi-source innovation but also allowed backwards compatibility that was required by physical stability and longevity of production equipment.

Second, through our analysis we show that increased digitizing of the platform functions resulted in an increasingly loosely coupled architecture and related dynamic growth patterns. This loose coupling promoted new stretch-fit patterns between increased technology variability and its alignment with use contexts, which calls for constant changes in firm's strategy and economic logic. As the platform gained increasingly strong digital capabilities and loosely coupled architecture, the functional scope was extended. This expansive feature, in combination with the strongly evolutionary characteristics of hybrid platforms, created dynamic growth patterns where technology, context or strategy stretched the platform's scope and evoked related alignment in the others.

A third contribution is that we demonstrate how generativity lead to increasingly distributed forms of control. As new layers were digitized, ABB's control over the platform shifted from full control of technical development and relationships to control of access. Enabled by digitizing the functions of the control system, and due to the expansive scope and intensity of learning associated with these changes, control became increasingly distributed. The fact that platforms and strategy co-evolved through a stretch-fit pattern as digital capabilities increased technology variation calls for new and increasingly frequent strategic alignments with use and production contexts.

These findings can be applied to a new breed of platforms emerging in such diverse fields as intelligent cities, power grids or new generation of traffic and power train systems. While hybrid platforms are an increasingly promising area for further leveraging the promises of digitization, our study also points to important challenges in their design and management. Nevertheless, this study has several important limitations. First, while pointing to important challenges our study cannot be generalized across all digitization of physical product platforms because we focused on an industry that is extremely dependent on high-reliability and has very long life-cycles of machinery. Accordingly, both evolutionary characteristics and dynamic growth patterns might be overemphasized. Second, our study examines evolution in a world-leading firm in a capital intensive high-tech engineering based industry. It is not clear how and under what conditions digitization of physical environments might play out in other contexts. Finally, industrial control systems are used in a large variety of industries and innovation adoption patterns might be quite different in other use contexts. Most of the larger customers discussed in this study operates in industries that have been highly profitable over the last decades due to rising prices of natural resource. Platform owners in more cost dependent industries might be subject to other dynamics.

6. Conclusion

Extensive research on platforms provides a thorough understanding on why they materialize, what their features are and how they provide value. However, we know less on how specific architectural principles, and in particular digital ones, affect their evolutionary trajectory.

In this paper we have shown how the evolutionary logics perspective allows for a deeper understanding of how hybrid platforms evolve as they become enabled by expansive digital capabilities. Although digitization has been raised as an important issue for research on new platforms (Tiwana et al., 2010) it has not been explored in detail. Overall, our study also illustrates challenges in designing and managing hybrid platforms that involve dynamic digital and physical components. By doing so our research makes several contributions.

First, our study extends current platform theory by illustrating how digitizing analog components created a new breed of hybrid platforms with strongly evolutionary characteristic and specific platform design challenges. Second, through our analysis we show that increased digitizing of the platform functions resulted in an increasingly loosely coupled architecture and related dynamic growth patterns. Finally, we demonstrate how generativity leads to increasingly distributed forms of platform control.

7. References

Baldwin, C. and K. B. Clark (2000). Design Rules: The Power of Modularity B2 - Design Rules: The Power of Modularity. Boston, Harvard Business School Press.

Baldwin, C. and J. Woodard (2009). The Architecture of Platforms: A Unified View. Platforms, markets and innovation. A. Gawer. Cheltenham UK, Edward Elgar: 19-44.

Baldwin, C. K. (2008). "Where do transactions come from? Modularity, transactions, and the boundaries of firms." Industrial and Corporate Change **17**(1): 155-195.

Baldwin, C. Y. and K. B. Clark (1997). "Managing in an age of modularity." Harvard Business Review **75**(5): 84.

Barrett, M., E. Oborn, et al. (2012). "Reconfiguring boundary relations: Robotic innovations in pharmacy work." Organization Science **23**(5): 1448-1466.

Benkler, Y. (2006). The wealth of networks: How social production transforms markets and freedom, Yale University Press.

Bosch, J. (2009). From software product lines to software ecosystems. Proceedings of the 13th International Software Product Line Conference.

Boudreau, K. (2010). "Open platform strategies and innovation: Granting access vs. devolving control." Management Science **56**(10): 1849-1872.

Boudreau, K. J. (2012). "Let a thousand flowers bloom? An early look at large numbers of software app developers and patterns of innovation." Organization Science **23**(5): 1409-1427.

Bresnahan, T. F. and S. Greenstein (1999). "Technological competition and the structure of the computer industry." The Journal of Industrial Economics **47**(1): 1-40.

Brusoni, S. and A. Prencipe (2009). "Design rules for platform leaders." Platforms, Markets and Innovation. Cheltenham, UK and Northampton, US: Edward Elgar.

CISCO. (2013). Retrieved February 6th, 2013, from http://www.cisco.com/en/US/docs/solutions/Verticals/CPwE/CPwE_chapter2.html - wp1002608.

Desouza, K. C. and A. Bhagwatwar (2012). "Citizen Apps to Solve Complex Urban Problems." Journal of Urban Technology **19**(3): 107-136.

Eaton, B., S. Elaluf-Calderwood, et al. (2011). Dynamic structures of control and generativity in digital ecosystem service innovation: the cases of the Apple and Google mobile app stores, London School of Economics and Political Science.

Eisenhardt, K. (1989). "Building Theories from Case Study Research." Academy of Management Review: 532-550.

Eisenmann, T., G. Parker, et al. (2006). "Strategies for two-sided markets." Harvard Business Review **84**(10): 92.

Evans, D. (2009). "How catalysts ignite: The economics of platform-based start-ups." Cheltenham, UK and Northampton, MA, US: Edward Elgar.

Evans, D. D. S., A. Hagiu, et al. (2006). Invisible engines, MIT Press.
Gao, L. S. and B. Iyer (2006). "Analyzing complementarities using software stacks for software industry acquisitions." Journal of Management Information Systems **23**(2): 119-147.
Gersick, C. J. G. (1991). Revolutionary Change Theories: A Multilevel Exploration of the Punctuated Equilibrium Paradigm. Academy of Management Review. **16:** 10-36.
Hanseth, O. and K. Lyytinen (2010). "Design theory for dynamic complexity in information infrastructures: the case of building internet." Journal of Information Technology **25**(1): 1-19.
Helper, S. and M. Sako (2010). "Management innovation in supply chain: appreciating Chandler in the twenty-first century." Industrial and Corporate Change **19**(2): 399-429.
Helpman, E. (1998). General purpose technologies and economic growth, The MIT press.
Henderson, R. M. and K. B. Clark (1990). Architectural innovation: the reconfiguration of existing product technologies and the failure of established firms. Administrative Science Quarterly. **35:** 9-30.
Kirschner, M. and J. Gerhart (1998). "Evolvability." Proceedings of the National Academy of Sciences **95**(15): 8420-8427.
Kjaer, A. P. (2003). "The integration of business and production processes." Control Systems, IEEE **23**(6): 50-58.
Lee, J. and N. Berente (2012). "Digital Innovation and the Division of Innovative Labor: Digital Controls in the Automotive Industry." Organization Science **23**(5): 1428-1447.
Majchrzak, A. and L. Markus (2012). Technology Affordances and Constraint Theory of MIS. Thousand Oaks, CA, Sage.
Meyer, M. H. and A. P. Lehnerd (1997). The power of product platforms, Free Press.
Remneland-Wikhamn, B. r., J. Ljungberg, et al. (2011). "Open innovation, generativity and the supplier as peer: The case of iphone and android." International Journal of Innovation Management **15**(01): 205-230.
Robertson, D. and K. Ulrich (1993). "Planning for Product Platlorms." Sloan Management Review.
Rochet, J. C. and J. Tirole (2003). "Platform competition in two,Äêsided markets." Journal of the European Economic Association **1**(4): 990-1029.
Sako, M. (2009). "10. Outsourcing of tasks and outsourcing of assets: evidence from automotive supplier parks in Brazil." Platforms, Markets and Innovation: 251.
Sanderson, S. and M. Uzumeri (1995). "Managing product families: The case of the Sony Walkman." Research Policy **24**(5): 761-782.
Simon, H. A. (1972). The Architecture of Complexity. Proceedings of the American Philosophical Society**:** 467-482.
Simon, H. A. (1996). The Sciences of the Artificial B2 - The Sciences of the Artificial. Cambridge, MA, MIT Press. **3**.
Tilson, D., K. Lyytinen, et al. (2010). "Research Commentary--Digital Infrastructures: The Missing IS Research Agenda." Information Systems Research.

Tiwana, A., B. Konsynski, et al. (2010). "Platform Evolution: Coevolution of Platform Architecture, Governance and Evironmental Dynamics." Information Systems Research **21**(4): 675-687.

Yoo, Y., R. J. Boland, et al. (2012). "Organizing for Innovation in the Digitized World." Organization Science **23**(5): 1398-1408.

Yoo, Y., O. Henfridsson, et al. (2010). "Research commentary: The new organizing logic of digital innovation: an agenda for information systems research." Information Systems Research **21**(4): 724-735.

Zittrain, J. L. (2006). "The generative internet." Harvard Law Review: 1974-2040.

Appendix A. Summary Tables

Table 4. The Master Product Family

Platform Aspect	Capability	
Information Integration Capability	N/A	
Engineering Approach	Product development	
Economic Logic	Re-use of physical engineering solutions, standardization of interfaces at the control-process level	
Operating System	In-house developed	
Platform Layers	5 Enterprise Network	NA
	4 Site business planning	NA
	3 Site Manufacturing Operations and Control	NA
	2 Area Supervisory Control	Basic text based operator stations integrated with a limited part of the process.
	1 Basic Process Control, Level 1	PLC with 16/32 bit processor introduced, provided improved process control and simplified commission and maintenance through standardization of physical engineering (replacing hardwired logic) a single controller had a capacity for 1300 channels
	0 Physical/Process Level 0:	Serial Communication Bus Introduced
Evolutionary Stretch	Strategy	First steps towards purposeful re-use through digitization of existing layers.
	Context/ functionality	The new technology was designed to fit the existing regime.
	Technology	Digital technology introduced.

Table 5. The Advant Platform

<table>
<tr><th>Platform Aspect</th><th colspan="2">Capability</th></tr>
<tr><td>Information Integration Capability</td><td colspan="2">Standards such as TCP/IP, SQL, DDE, and X Window System allows information to be retrieved. Integration approach based on information that was stored centrally, i.e. the system managed data storing for different applications.</td></tr>
<tr><td>Engineering Approach</td><td colspan="2">Product development but inspired by modular software logics such as object oriented approach, integration of information</td></tr>
<tr><td>Economic Logic</td><td colspan="2">Re-use of both in-house developed software solutions and externally developed technology through standardization and integration of engineering systems. Modular strategy of product development aims at providing system expandability. Support for both on-line configuration and off-line engineering.</td></tr>
<tr><td>Operating System</td><td colspan="2">Unix</td></tr>
<tr><td rowspan="6">Platform Layers</td><td>5 Enterprise Network</td><td>NA</td></tr>
<tr><td>4 Site business planning</td><td>NA</td></tr>
<tr><td>3 Site Manufacturing Operations and Control</td><td>Support for a range of interfaces and protocols for communication with external systems such as communication modules for direct connection with third party control equipement, EXCOM, TCP/IP, GCOM and SQL*Connect.</td></tr>
<tr><td>2 Area Supervisory Control</td><td>Modules ranging from low-cost, monochrome video terminals, to advanced operator with high-resolution color graphics and windowing based on industry standards, such as X Window System and OSF/Motif.</td></tr>
<tr><td>1 Basic Process Control,</td><td>The AC 450 had a substantially increased computational capability and provided redundancy. Process control could be customized to customer needs through a range of controllers, varying from small PLC units for a maximum of 64 I/O channels to advanced controllers for logic and regulatory control with as many as 5700 I/O channels per station.</td></tr>
<tr><td>0 Physical/ Process</td><td>Fieldbuses for I/O units, operator stations, local interaction panels and controllers</td></tr>
<tr><td rowspan="3">Evolutionary Stretch</td><td>Strategy</td><td>Increased reliance on third-party modules changes ABB’s possibilities to control development. Openness and integration becomes increasingly important in the platform value proposition.</td></tr>
<tr><td>Context/ functionality</td><td>The platform expands into layer 3 and provides substantial new functionality in Layer 2. New software for engineering provides functionality for platform design and customization.</td></tr>
<tr><td>Technology</td><td>Leveraging general-purpose digital technology, the architecture becomes increasingly layered and loosely coupled. For good and worse, generativity becomes a factor.</td></tr>
</table>

Table 6 The Advant Integrator Platform

<table>
<tr><th>Platform Aspect</th><th colspan="2">Capability</th></tr>
<tr><td>Information Integration Capability</td><td colspan="2">Integration approach based on applications, through well-defined application programming interfaces (API's)</td></tr>
<tr><td>Engineering Approach</td><td colspan="2">Software platform</td></tr>
<tr><td>Economic Logic</td><td colspan="2">Varies over time; product platform, market platform</td></tr>
<tr><td>Operating System</td><td colspan="2">Windows</td></tr>
<tr><td rowspan="6">Platform Layers</td><td>5 Enterprise Network:</td><td>Integration capabilities towards large ERP systems and other existing digital assets</td></tr>
<tr><td>4 Site business planning</td><td></td></tr>
<tr><td>3 Site Manufacturing Operations and Control</td><td></td></tr>
<tr><td>2 Area Supervisory Control</td><td></td></tr>
<tr><td>1 Basic Process Control</td><td>Computational capability vastly improved, overall functional improvements are however incremental since these changes are not followed by equivalent investments in physical layers.</td></tr>
<tr><td>0 Physical/
Process</td><td></td></tr>
<tr><td rowspan="2">Evolutionary Stretch</td><td>Strategy</td><td>The platforms fundamental purpose is to integrate layers in the loosely coupled, layered system. As such, the core value proposition is not limited to developing the best solutions but rather to control and certify software and hardware modules. At first, management does not consider the platform to be restricted to controlling industrial processes. Strategic circumstances in E & A does however limit operations to core functions.</td></tr>
<tr><td>Context/
functionality</td><td>As the upper layers of the platform are digitalized, functionality expands into integration of general-purpose digital technology. Physical restrictions and longevity of production equipment restricts fundamental changes in the lower layers. Management explores expanding platform use context into other industries but</td></tr>
</table>

		strategic considerations interrupts such stretch patterns.
	Technology	Integration capabilities improved and interfaces towards higher layers improved. As platform is based on Windows, E & A does not have decision rights over development either for OS or many other modules. In order to adapt to changes in these components the platform becomes increasingly dynamic and re-programming of digital components more frequent.

Table 7, The 800XA Platform

Platform Aspect	Capability	
Information Integration Capability	Integration approach based on applications, through well-defined application programming interfaces (API's). Platform designed for integration of in-house and third-part developed modules.	
Engineering Approach	Software eco-system platform	
Economic Logic	Varies over time; market platform, product platform, service platform	
Operating System	Windows	
Platform Layers	5 Enterprise Network:	Further improved integration towards important systems, e.g. large ERP system maintenance module
	4 Site business planning	Improved functionality in the platform and more powerful integration with third party systems.
	3 Site Manufacturing Operations and Control	New focal point of the process automation system.
	2 Area Supervisory Control	Smaller systems (up to 10 000 objects) can be implemented with server functionality directly on workplace nodes. Larger systems runs on customized servers, functionality. While process control is still managed by designated controllers in level 1, some acyclic, service and configuration related data is bypassed (increased loose coupling)
	1 Basic Process Control, Level 1:	Computational capability vastly improved, overall functional improvements are however incremental since these changes are not followed by equivalent investments in physical

		layers.
	0 Physical/Process	N/A
Evolutionary Stretch	Strategy	Commoditization of IT modules lowers margins on products, service and gatekeeper role increasingly important for ABB
	Context/ functionality	Physical restrictions imply that benefit of IT innovation in the platform is limited, changes seen as risk. At the same time, users are becoming more IT savy end expects to be able to integrate both hardware modules such as video cameras and software such as ERP systems' maintenance modules
	Technology	The digital components are now built on standardized, commercial off-the-shelf modules. As such, they are subject to innovation dynamics in the general IT industry.