This is a preprint of a paper accepted for publication in Nordic Contributions in IS Research and presented in the 7th Scandinavian Conference of Information Systems. The final publication is available at Springer Link: http://link.springer.com/chapter/10.1007/978-3-319-43597-8_5

The Ambivalent Characteristics of Connected, Digitised Products: Case Tesla Model S

Antti K. Lyyra, and Kari M. Koskinen

Department of Management, The London School of Economics and Political Science, London, United Kingdom
{a.k.lyyra,k.m.koskinen}@lse.ac.uk

Abstract. Connected, digitised products are assemblages that comprise digital and physical components and are linked to digital support infrastructures. Given that re-programmability of digital components allows a product designer to adopt a design philosophy that embraces incompleteness and continuous improvement, digitised (tangible) products may also become incomplete and open-ended if such a philosophy is embraced. This research uses Tesla Model S as a case study to explore the mutability of a passenger car over time. The results show that a type of a product that has traditionally been seen as stable may become open-ended, incomplete and mutable in terms of its specifications and functionality. This brings forward the relevance of complementary architectural frames and principles to conceptualise differing design cycles among physical and digital components in innovation and product management while also showing blurring boundaries of control between the owner and the manufacturer of a product.

Keywords: connected, digitised products, hybrid materiality, design cycles, product lifecycle, innovation management, product management

1 Introduction

Innovation and product management practices strive to connect organisational capabilities to customer demands [12] over product lifecycles, ranging from design, manufacturing and maintenance to decommissioning and disposal.

Traditionally, production is perceived as a compilation of components to produce a final product [11]. That product is in turn perceived as a complete and fixed output that is sold to a customer. Whenever innovation occurs, it almost invariably leads to the design and introduction of a new model. Therefore, product design is seen as a clear and somewhat separable cycle, which is followed by production and eventually leading to the sales and transfer of ownership. After that, apart from a maintenance type of repair or service, products are not generally expected to improve in specifications or receive any new functionality.

The digitalisation of tangible products is challenging this view. While the physical components of products resist change, the digital components are more open-ended and readily changeable [5, 9, 17]. Therefore, the incorporation of digital components into physical tangible products blurs the boundaries of to what extent a product is perceived as amenable to change, or according to which criteria a particular product could be seen as finished or complete. Moreover, added with connectivity to digital infrastructures and cloud-based support infrastructures, it is no longer clear where the boundaries and control of a product reside.

This study sets out to explore how such open-endedness of digitised tangible products might impact innovation and product management practices as well as customer experience using the Tesla Model S passenger car as a case study. The topic is approached by examining software releases that introduce changes to specifications and functionality of the car. This case is seen as an outlier in an industry that traditionally has focused its efforts on the design and production of well-engineered and complete products.

The results show that a product, which is traditionally seen as stable, can become open-ended and incomplete. The various specifications and functionalities are constantly changing, showing what it means in practice when a tangible product becomes increasingly digital, mutable, connected and dependent on its manufacturer throughout its lifetime. The results also empirically confirm the relevance of utilising complementary architectural design principles to conceptualise differing design cycles and levels of resistance to change among digital and physical components over the lifecycle of a product [7].

The paper is organised as follows. Chapter two discusses characteristics of hybrid materiality of digitised products and outlines potential impacts of this materiality in innovation and product management practice, after which chapter three presents the research question to explore empirically how a product of hybrid materiality changes over time. Subsequently, chapter four presents the case and summarises data collection and analysis procedures and methods. The results of the analysis are presented in chapter five, whereas chapter six discusses the findings and their meaning together with limitations and possible avenues for future research. Chapter seven concludes this paper.

2 Literature Review

The success of commercial enterprises commonly depends on their ability to introduce new products in a fashion that brings novelty and satisfaction to their customers. While product innovation has traditionally followed relatively predictable design cycles, increased use of digital components in tangible products is poised to alter the frequency of design iterations.

The material characteristics of physical and digital components in terms of their amenability to change have gathered interest among digital innovation researchers. Broadly speaking, the physical objects, components and artefacts are described as resistant to change; the digital objects, such as computer programs, are described as less resistant, something that can be separated from their physical carriers [13] and moulded from one form to another with relative ease [7, 17].

This lack of resistance in the digital domain is said to follow from the re-programmability of digital computers and homogenisation of data [13, 17]. The digital computer is a general purpose machine, which is able to process any set of step-wise instructions (software) as long as such a set is presented to a computer in an agreed and unambiguous format of binary numbers (bits). The notions of re-programmability and homogenisation of data imply that behaviour of the digital computer is readily changeable; it can be repurposed with minimal marginal cost and delay by supplying it with new sets of instructions [7]. As the functioning logic of a computer can be changed with relative ease, the digital materiality is often described as something that is malleable and amenable to change [9].

In contrast, the physical artefacts and objects typically lack ways for cost effective and instant collective changes. More tools, time and labour are needed to carry out modifications for each unit, increasing the marginal costs and making them more resistant to attempts of change and repurposing.

The concept of digitised (tangible) products has been used to describe products, which are assemblages of the digital and physical components [7], whereas the concept of hybrid materiality [2] has been used to describe the material characteristics of products that combine digital and mechanical components with asymmetric resistance to change.

Given the changeability of digital materiality, functionality and behaviour of digitised products can be redesigned and changed several times after their first introduction. By doing so, they can be made to better-match circumstances and environments, which might not have been even foreseen during the original design phase, yet pose no real problem as adaptations to this type of products can be made at any point during their lifecycle. As a result, similarly to software, digitised products can also be seen as unfinished, malleable and ontologically ambivalent [5, 9, 18].

Consequently, it appears that there is no urgent need to restrict the design and development of a digitised product to any particular point in time. Existing functionalities can be changed and new ones introduced continuously as long as they are in the reach of digital means and there is a connection to provide a product with a new set of software. While software can be easily changed, it is important to note that the logic embedded in software must conform with hardware as well as the aspects of the physical environment it seeks to model and manipulate. In that sense, digital and physical materiality are tightly connected as the physical restrains the digital; it can be argued that it places boundaries on what can or cannot be achieved by changing the digital [8].

The notion of abstract design patterns as a design principle has been proposed as a useful addition to conceptualise and manage the tensions that may emerge from the differing design cycles and levels of resistance among physical and digital components [7]. This forwards an idea of conceptualising design not simply as a hierarchy-of-parts, but rather as something which is combined with networks-of-patterns; well-defined and specified plans and components are complemented with generalisations, which are abstracted patterns that represent intended functions or purposes of components. Patterns are seen as comparable to placeholders for something that can be invented or improved later. The applicability of this principle is demonstrated in the case study of infotainment architecture design and development at a car manufacturer [7]. Although the authors of that study theorise that digitised products are amenable for change throughout their lifetimes, their empirical evidence does not reach beyond the design phase and is limited to the components and systems that revolve around an infotainment system.

The incomplete nature of a digitised product also begs the question on who decides when and how to change the functioning of a product. The incompleteness of a digitised product and its linking to a cloud based system have potential to shift some parts of control from the owner of a product to its manufacturer. Furthermore, the latter could introduce a change in specifications and functionalities of a product without seeking the approval of its owner. This deviates from the notion where the control of a product has resided almost entirely in the hands of its owner. In terms of seeing digitised products sharing characteristics with more traditional technological platforms, the tendency there has been for the platform owner to try to hold most of the control of the platform [3, 13].

As shown above, the digital and physical materiality differ in their capacity and resistance to change. Whereas the physical materiality is harder to change, the digital materiality portrays itself as open-ended that can changed repeatedly. While this ambivalent ontology of digitised products may provide a broad sphere of design and marketing choices for innovation and product management over a lifetime of a product, it also raises questions regarding consumer experience and where the ultimate control of a particular product actually resides. To explore the avenues of hybrid materiality and abstract design patterns, this study sets out to explore to what extent a product that is traditionally seen as stable can be changed through digital means after its production when it is already in use.

3 Research Question

To further investigate how this ambivalent ontology of digitised products might impact on innovation and product management strategies and practices, this study sets out to investigate the frequency and dimensions mutability of a passenger car once it has rolled out of the factory.

Typically, new car models are brought to market every four to seven years and supplemented with cosmetic facelift models about halfway through a model's lifecycle. A car is expected to stay much the same once it has been manufactured; oil, spark plugs and other consumables are changed and software updated in a service, but the specifications and functionalities are not expected to change. In other words, they are purchased as finished and complete products with their functionality fully specified and known.

This lifecycle model is being challenged. The recently established car maker Tesla sends out frequent software releases to change and improve functionality of the cars they have manufactured and sold while other car makers are planning to introduce the possibility of updating software of cars over the Internet connection. This introduces a novel time-based dimension to the configuration space of specifications and functionalities of digitised products.

To explore further the frequency of change and the functional domains that could benefit from the utilisation of abstract design patterns, we approached the topic through the following research question:

What are the characteristics of digitally-introduced change in a connected, digitised product?

The aim is to answer the question in three ways by surveying reported software releases with their respective functionality of the Tesla Model S car. First, the frequency of digitally introduced change will be established by surveying the quantity and quality of software releases. The physical changes and recalls are not in the focus of this analysis. Second, the contents of the software releases are examined and grouped in order to understand how specifications and functionalities of the car change over time. These groups demonstrate the scope of domains amenable to abstract design patters. Third, other relevant qualitative aspects of change are collected from the data and analysed. These include, among others, shortcomings found from the software, hardware-software conflicts and contractual and regulatory restrictions instantiated through software.

4 Case Selection

We focus on a particular model of Tesla, the Model S, in order to form a coherent understanding on what kind of changes can be and are made for a particular model through software updates. The customer deliveries of the Model S started in June 2012 and it has been on the market for more than three years now, going through several software releases and therefore providing longitudinal empirical evidence for this study. The Model S can be seen as a typical example of a passenger car in the sense that it is meant to be used in average road conditions and it is not meant for off-road activities or any other kind of special circumstances.

From a case study perspective, the Model S can be seen both as a representational and as an outlier case [15]. It is representational since it belongs to a rather common and well-established product category group of passenger cars. On the other hand, within this category it is also an outlier. What sets Tesla apart from the other car manufacturers is its heavy exploitation of open-endedness of connected, digital materiality. This happens not only in terms of the control functions of the car, such as adjusting the functioning of energy and power-train, but also in providing constant cloud-based software updates that can change specifications and functionality of the car in various ways. Much of the hardware of the model has been designed to offer maximum flexibility for software updates. As an example, the control panel of the car is a 17 inch touchscreen with no physical buttons, which enables it to be not only completely editable in relation to what it shows, but also in how it works.

It is expected that the case provides a useful empirical starting position to discuss further the potential implications of digitalisation to innovation and product management practices and consumer experience. Although much of mutability of a digitised product is enabled by the ambivalent characteristics of a digitised product, it is important to note that mutability does not automatically follow from any digital materiality embedded in a product. It should rather be seen as a design strategy or decision that is enabled by the hybrid materiality of digitised products. To prove the point, most of the modern cars are seen as stable products although they contain and exploit a large amount of software, yet they are not constantly in the making and are not generally expected to change over their lifetime.

4.1 Data, Method and Analysis

To update cars, Tesla delivers software releases in a phased manner; once an update is available, a car displays a notification on the screen with an option to install it immediately or to schedule the installation for a later time. Tesla's software release procedure contains major, minor and maintenance type of releases. A software version can be identified either by a release number or a software build number, both of which are shown at the time of update. Typically, major releases are issued with release notes whereas minor releases occasionally lack them. Maintenance releases seem to be left unspecified. Given that the authors did not have direct contact with Tesla, and the company does not make software release histories and their specific contents publicly available, the empirical evidence was gathered and combined from release notes that were available on the Internet, the Wiki and discussion sections of Tesla Motors Club[1] and Tesla Motors[2] and Teslarati[3] websites. All the data was collected online, following the example of other studies that have been able to show how online content can function as a valuable source of data for research [3, 14]. The evidence was arranged by listing software releases with their respective release and software build version numbers and dates and then matching respective releases with changed functionalities for further analysis.

The analysis started by surveying the frequency of software releases and calculating the average amount of days between releases. After establishing the frequency of releases, the qualitative data analysis was performed using the methods of thematic analysis [1]. The coding process started by creating categories based on the open coding of a subset of data. Subsequently, the analysis continued by examining changes delivered to cars through software releases over a period of time. The changes were assigned to categories according to their functional features. The evidence was also analysed according to the notions of restrictions and quality. Restrictions were mostly subject to two different factors: geographical location and configuration of a car. Quality refers to notions of improvement and decline in quality that are related to shortcomings in software and to the corrections of such shortcomings.

[1] http://www.teslamotorsclub.com
[2] http://www.teslamotors.com
[3] http://www.teslarati.com

The data was coded by the two authors individually in order to guarantee the overall quality and inter-coder agreement. The agreement between coders was approximately 90 percent, and where the authors did not agree, discussion followed as to why not. The disagreements on the coding were mainly due to functionalities that affected different functional areas of the car and it was not clear which one of the functional areas prevailed over the others.

5 The Results of the Analysis

This chapter presents the results of the analysis. To begin, the frequency of software releases demonstrates the rate of mutability of a digitised product. This is followed by outlining the domains of functionalities and specifications that are amenable to the open-ended principles of abstract design patterns. Finally, restrictive and qualitative aspects of software changes are brought forward before closing remarks.

5.1 The Frequency of Change

Tesla has provided the Model S with five branches of major release over the past three and half years (Table 1.). The first branch I(*) was released when the first Model S was delivered to a customer on 22 June 2012, whereas the current major release 7 was first pushed to cars on 16 October 2015. The time between moving from one major branch to the next one varied from 161 to 431 days. During the lifecycles of major branches, each of them received 1 to 10 minor releases and numerous maintenance releases. In total, over this period of time, Tesla has issued 117 software releases for the Model S, approximate one every 11.49 days.

5.2 Changes in Specifications and Functionalities

The analysis of the changes introduced by software releases established seven functional domains that were amenable to the open-ended principles of abstract design patterns, namely: information, entertainment, user interface design, energy and performance, ancillaries, connectivity and self-steering capabilities. The first three categories are often subsumed under a higher-level category of infotainment [7], however, we treat the three categories as separate for analytical purposes. It is also worth noting that these categories are not mutually exclusive; often a single feature or a change within a release could span across several categories.

The *information* category is comprised of changes to driving related map and navigation functionalities, as well as to personal time management and communication applications. The maps and navigation updates revolved around traffic information and functionalities, often with an aim to better localise, plan a route through and navigate to charging stations. The changes in time management and communication applications were related to functionalities to connect a driver's phone with a car's information systems using the Tesla mobile application.

Table 1. Software releases with their respective frequency

Major Release Branch	**I (*)**	**4**	**5**	**6**	**7**	**All**
Releases in Branch	1.7.36 1.9.11 1.9.17 1.13.16 1.15.8 1.15.14	4.0 4.1 4.2 4.3 4.4 4.5	5.0 5.5 5.6 5.8 5.9 5.10 5.11 5.12 5.14	6.0 6.1 6.2	7.0 7.1	All
Lifetime in Days (approx.)	161	232	431	375	133	**1344**
First Release	22 Jun 2012 (**)	30 Nov 2012	18 Jul 2013	11 Sep 2014	16 Oct 2015	22 Jun 2012
Latest Release	30 Nov 2012	20 Jul 2013	22 Sep 2014	21 Sep 2015	26 Feb 2016 (***)	26 Feb 2016 (***)
Total Number of Software Builds	6	19	32	45	15	**117**
Number of Major Releases	1	1	1	1	1	5
Number of Minor Releases	5	5	10	2	1	23
Number of Maintenance Releases	0	13	21	42	13	89
Average Number of Days between Software Releases	**26.83**	**12.21**	**13.47**	**8.33**	**8.87**	**11.49**

* The first major branch is referred to as I for initial.
** The first customer delivery of the Model S is used as the starting date.
*** The latest release to this active branch is used as the end date.

The *entertainment* category includes changes that are geared towards entertainment, such as the introduction of the Spotify streaming service as well as enhancing the ability to browse a USB stick for media content. Improvements in radio buffering and reception and web browser performance and stability were considered to represent the categories of information and entertainment simultaneously.

The *user interface design* category was formed on the basis of updates in information and entertainment functions that were more of a visual than practical characteristic. Such changes include new voice commands, the introduction of "flat looks" and changes in font size and auto-brightness and search-box functionalities. Also, the changes to the behaviour of control buttons on the steering wheel are considered to be represented by this category.

The *energy and performance* category represents the chances and improvements in mileage, acceleration and top speed, the changes that revolve around battery management and the control of electric motors. Mileage maximising functionalities intend to minimise something that is often referred to as “end of range anxiety” (a fear of a car running out of battery), whereas acceleration and top speed are more geared towards maximising the driving experience.

The *ancillaries* category characterises changes to software that controls ancillary functions such as locks, door handles and windscreen wipers, introducing chances to their behaviour.

The *connectivity* category emerged to describe the changes that were introduced to improve a car's connectivity to telecommunications infrastructure and cloud-based services over 3G and Wi-Fi protocols.

The *self-steering* category contains updates that change the auto-steering capabilities and therefore alter the behaviour of a car. Such capabilities include traffic-aware cruise control, lane keeping with automatic steering, self-parking and automatic high/low beam headlights, which are a part of an autopilot convenience features package as well as collision warning systems, all of which allow a driver to shift some driving-related tasks to a car.

On average, each release led to the introduction or changing of 6.6 control setting items in the control panel, based on calculations from the release notes. Also, the analysis of functions showed that the emphasis of earlier releases were on the side of basic functionality and incremental enhancements, while the latter ones were more geared towards introducing new functionality.

5.3 The Restrictive and Qualitative Aspects of Change

The analysis revealed also that changes were not only about positive improvements. While generally aimed at improving and enhancing functionality, in some occasions bugs were also introduced. Wi-Fi issues, touch screen flashing, rebooting to overcome issues as well as unexpected shutdowns were reported. Also, while Tesla was investigating the causes of battery fires in late 2013, it removed active air suspension control to raise the ride-height and increase road clearance for precautionary and safety reasons. Once the investigation was closed and the protection of batteries improved by retrofitting stronger battery shields, the suspension control and lower ride-heights were reintroduced with a later software release.

Conflicts and interdependencies among hardware and software configurations were also encountered. To provide examples, the release of version 5.0 introduced GPS errors in certain cars, which prompted Tesla to replace hardware in the affected cars. Also, certain auto-steering capabilities are not available to the cars that lack appropriate hardware such as sensors. There were interdependencies among software-based configurations as well. If a certain software package was not purchased for a car, even if that car had the appropriate set of hardware installed, certain specifications and functions from a release were not available until that package was purchased for a car.

Furthermore, Tesla provided mobile applications to remotely monitor and control a car only for iPhones and Android, leaving Windows Phone and BlackBerry users without a possibility to control their cars through mobile phones.

Another set of boundaries was found relating to geographical, contractual and regulatory reasons. The music streaming service Spotify was introduced only to certain geographical areas, whereas other areas were served by other music streaming services, and certain auto-steering functions are available only in certain countries. In Hong Kong, after Hong Kong's Transport Department requested a review of safety measures, Tesla temporarily disabled the Autosteer and Auto Lane Change functions on all Model S in Hong Kong with immediate effect.

The results show that the hybrid materiality of the car allows it to be changed frequently through digital means while simultaneously revealing functional domains that can be subjected to the open-ended principles of abstract design pattern philosophies. This multidimensional mutability characterises a product that is connected to centralised cloud-based infrastructures, incomplete and constantly in the making.

6 Discussion

The analysis of empirical evidence shows that a type of product that has traditionally been seen as relatively stable may become open-ended, incomplete and mutable within several functional and qualitative dimensions, depending on a particular configuration of hardware, software and connectivity. This demonstrates the applicability of digital theories in product-based domains and shows that the notion of abstract design patterns could be applied to conceptualise a wide range of mutability of digitised products throughout their lifecycles. The implications of mutability are envisaged to have impact on innovation and product management practices and consumer experience while boundaries of control are shifting.

To begin, the analysis confirms our initial, literature-driven conception, which shows that digitised products that consist of hybrid materiality may become ambivalent in their specifications and functionality. Given that much of that continuous change is enabled by the re-programmability of digital computers and digitalisation, connectivity, platforms and cloud-based infrastructures as general phenomena, it is envisaged that theories of digitalisation and digital innovation could gain more relevance outside typical domains of information systems research community. This supports the argument that digital innovation should be considered as a fundamental and powerful concept in the information systems curriculum [4] and shows that the lessons from digital innovation could contribute to innovation and product management research [16].

The mutability that is enabled by the digital materiality, however, is constrained by the physical materiality that sets boundaries on what functionalities and specifications can and cannot be modified through the means of releasing computer software. This tension among materialities was demonstrated in the study that examined differing design and development cycles among physical and digital components during the design phase of an infotainment systems of a passenger car [7]. Consequently, they introduced the notion of architectural frames and proposed the combination of types of frames, hierarchy-of-parts and network-of-patterns as a complementary ways to frame and conceptualise the rigidity of physical materiality and the open-endedness of digital materiality. The results of this study demonstrate empirically that the above-mentioned dynamics are not only pertinent to the design phase but may continue be exploited over the entire lifecycle of a digitised product. Also, abstract and mutable design patterns can be exploited along many dimensions of specifications and functionalities, ranging from infotainment to auto-steering capabilities, which are supported by cloud-based digital infrastructures and connectivity. The results imply a potentially increasing emphasis on developing digital-physical architectural frames that support changes in specification and functionality of a digitised product throughout its entire lifetime.

Utilising the complementary concept of architectural frames to make sense of an entire lifecycle from design to decommission suggests novel avenues and challenges for innovation and product management practices. On one hand, companies would have a larger spectrum of choice in their product and marketing decisions. To lower the upfront costs of product design, a company could decide to enter the market with a product that is good enough to get traction and subsequently improve and maintain it until decommissioning, should a need emerge either from the customer or the environment [5]. On the other hand, marketing messages could be tailored to emphasise the promise of continuous innovation and improvement instead of fine-tuned and complete products. This can give choices to consumers as well. They could choose to purchase a complete product that is fully known at the point of purchase, or, alternatively, to subscribe to a stream of improvements as well as an occasional decline, to release notes and the learning of new features, specifications, and settings. In order to bring market needs and organisational capabilities together [12], innovation and product managers and strategists should plan for mutability of a product throughout its lifecycle by reflecting upon the configuration of architectural frames [7], innovation [6] and cloud-based infrastructures that are required to support it [13].

While opening the spectrum of choice, the mutability of a connected, digitised product also blurs the limits of control between product owners and manufacturers. After the purchase it has traditionally been a matter for the owner to decide how the product he or she owns is going to change. With connected and digitised products, this aspect may no longer be valid. Since the manufacturer can alter a product, it also retains a significant portion of control over the specifications and functionalities of a product. This might not be much of a problem as long the software releases are seen as improvements. However, as products get more complex and their parts more interconnected, the verdict on whether a particular software release is considered as an improvement or a deterioration becomes more difficult to make. The increased interconnectedness among the functions of a car also means that an update in one area is likely to affect the functioning of other areas as well. A software release may not only enable new functionality, but also change the way the already existing functionality works, which might not always be received positively by the owners, but instead with more mixed reviews. Furthermore, as software release cycles become potentially shorter, it also raises the question as to how much change an average user is willing to accept. These constant changes place the user in a continuous learning mode, which can affect the overall satisfaction with a product. Therefore, the connected, digitised products have the capacity to change the relationship between manufacturers and owners, and not necessarily to the liking of the latter.

This issue in the area of control resonates with other issues relating to the control aspects of the digital, such as platform ownership [3] and the regulating power of code [10]. However, the connected, digitised products offer a slightly different research domain on control, as the key theme is the shifting power balance between manufacturers and customers. In the case of Tesla, it seems that it is expected by the manufacturer that the users simply agree with and adapt to the updates. Further research is needed to explain the implications that this shift in control entails for customer relations, but also in other areas such as product ownership and the tuning processes that take place between the users and the manufacturers of connected, digitised products, also during the creation of the actual physical components of the product.

Overall, understanding the changes that are brought about by the connected, digitised products is of interest not only to practitioners but also to many scholars working in fields related to the topic. Even though this paper sheds further light on the questions at hand, it is admitted that generalisations based on a single case study are not very straightforward to make. Additional data and research is therefore needed not only to understand the implications connected, digitised products have in other industries and to their organisational strategies, but also to explore and theorise further on the hybrid materiality of digitised products as a whole. What remains clear is that the area provides several fruitful avenues for digital innovation and management researchers to test and enhance their theories and take them to new contexts and domains of application.

7 Conclusion

This paper provides an example of how innovation and product management philosophies of incompleteness and continuous improvement can be appropriated and exploited by an industry that has traditionally aimed at completely designed and finished products. The enabling factors behind the phenomena are the hybrid materiality of digitised products and the lifecycle-long connection between a product, its user and its manufacturer. Not cutting this digital cord between the product and its manufacturer has important implications to various areas of innovation and product management, ranging from product design to managing customer relations. More research is needed to better understand the effects that connected, digitised products can have in different industries and to various stakeholders. The authors of this work hope that it provides some indication on what are the factors future research could pay attention to.

Acknowledgements. This research was supported by research student grants that were made available by the Emil Aaltonen foundation and the KAUTE foundation (the Finnish Science Foundation for Economics and Technology).

References

1. Attride-Stirling, J.: Thematic networks: an analytic tool for qualitative research. Qualitative Research. 1, 3, 385–405 (2001).
2. Barrett, M. et al.: Reconfiguring Boundary Relations: Robotic Innovations in Pharmacy Work. Organization Science. 23, 5, 1448–1466 (2012).
3. Eaton, B. et al.: Distributed Tuning of Boundary Resources: The Case of Apple's iOS Service System. MISQ. 39, 1, 217–243 (2015).
4. Fichman, R.G. et al.: Digital innovation as a fundamental and powerful concept in the information systems curriculum. MISQ. 38, 2, 329–353 (2014).
5. Garud, R. et al.: Incomplete by design and designing for incompleteness. Organization Studies. 29, 3, 351-371 (2008).
6. Henderson, R.M., Clark, K.B.: Architectural innovation: the reconfiguration of existing product technologies and the failure of established firms. Administrative Science Quarterly. 35, 1, 9–30 (1990).
7. Henfridsson, O. et al.: Managing technological change in the digital age: the role of architectural frames. JIT. 29, 1, 27–43 (2014).
8. Jarrahi, M.H.: Digital and Physical Materiality of Information Technologies: The Case of Fitbit Activity Tracking Devices. Presented at the 2015 48th Hawaii International Conference on System Sciences (HICSS) (2015).
9. Kallinikos, J. et al.: The ambivalent ontology of digital artifacts. MISQ. 37, 2, 357–370 (2013).
10. Lessig, L.: Code - version 2.0. Basic Books, New York (2006).
11. Muffatto, M.: Introducing a platform strategy in product development. Int. J. of Production Economics. 60-61, 145–153 (1999).
12. Teece, D.J. et al.: Dynamic Capabilities and Strategic Management. Strat. Mgmt. J. 18, 7, 509–533 (1997).
13. Tilson, D. et al.: Research Commentary—Digital Infrastructures: The Missing IS Research Agenda. Information Systems Research. 21, 4, 748–759 (2010).
14. Vaast, E. et al.: Talking about Technology: The Emergence of a New Actor Category Through New Media. MISQ. 37, 4, 1069–1092 (2013).
15. Yin, R.K.: Case study research: design and methods; 4th ed. Sage, London (2009).
16. Yoo, Y.: The Tables Have Turned: How Can the Information Systems Field Contribute to Technology and Innovation Management Research? JAIS. 14, 5, 227–236 (2012).
17. Yoo, Y. et al.: Organizing for Innovation in the Digitized World. Organization Science. 23, 5, 1398–1408 (2012).
18. Zittrain, J.: The Future of the Internet--And How to Stop It. Yale University Press (2008).