Technische Universität Dresden
Medienzentrum

Prof. Dr. Thomas Köhler
Dr. Nina Kahnwald
(Hrsg.)

GeNeMe '12

Gemeinschaften in Neuen Medien

an der
Technischen Universität Dresden

mit Unterstützung der

BPS Bildungsportal Sachsen GmbH
Campus M21
Communardo Software GmbH
Dresden International University
Gesellschaft der Freunde und Förderer der TU Dresden e.V.
Hochschule für Telekom Leipzig
IBM Deutschland
itsax - pludoni GmbH
Kontext E GmbH
Medienzentrum, TU Dresden
Webdesign Meier
SAP AG, SAP Research
T-Systems Multimedia Solutions GmbH

am 04. und 05. Oktober 2012 in Dresden

www.geneme.de
info@geneme.de

B.3 Open Innovation by Opening Embedded Systems

Constantin Söldner, Frank Danzinger, Angela Roth, Kathrin Möslein
Information System I – Innovation and Value Creation
University of Erlangen-Nuremberg

1 Introduction

With the increasing capabilities of today's smart phones, the demand of consumers for new applications has risen dramatically. By opening up these smart phones and providing third parties the opportunity to develop "apps" for their systems, producers like Apple and platform owners like Google can offer much more value to their customers. As smart phones are one kind of embedded systems (ES), the question arises if similar development can also take place in other kinds of embedded systems. ES, consisting of hardware and software, are embedded in a device to realize a specific function, in contrast to personal computers, which serve multiple purposes [4,30]. The notion of incorporating external actors in the innovation process has been coined open innovation which has become increasingly popular in research and practice since Chesbrough introduced the term in 2003 [12]. By opening up their innovation processes for external actors, firms could benefit from internal as well as from external ideas. In this paper, the notion of open innovation will be explored in the context of ES. The case of ES is particularly interesting, as it requires not only the opening of innovation processes, but also the opening of the embedded system itself. Some of these platforms are opened only to a small degree like Apple's iPhone, in order to enable others to create new applications for it. Similar developments also take place for example in the automotive software domain, especially concerning infotainment systems. However, most kinds of ES have been spared out by this development until now. As more than 98% of all chips manufactured are used for ES [10] and high-performing computer chips are getting cheaper [38], opening considerations could also prove valuable for a large number of other application domains. However, opening up innovation processes in the context of ES is challenging from both an organizational and technical perspective. First of all, embedded systems are subject to a variety of constraints in contrast to multi-purpose computing devices, like real-time and security constraints or costs and resource constraints. Second, ES are quite diverse both in their composition and in terms on their requirements. In this paper, we want to explore, how the different properties of embedded systems influence possible open innovation processes. This will be done by drawing on to the characteristics of firms implementing the three core open innovation processes suggested by Gassmann and Enkel (2004) [15] and conceptually explaining how the characteristics of ES enable or hinder open innovation processes. As a result, a classification of the OI processes in terms of ES characteristics is provided.

2 Openness of technical systems

In this section, motivations to open technical systems will be explored, in particular with respect to software platforms. The notion of openness has been explored in various fields. Regarding technologies, openness "relates to the easing of restrictions on the use, development, and commercialization of a technology" [7]. One motivation for opening a technology is for instance to increase the diffusion of the technology, as opening reduces the dependency of adopters to a single vendor [6,37]. By opening some of the components to enable external participation in development, openness is especially beneficial for systems consisting of multiple components [7]. Other advantages of opening are "the improvement of individual components; the creation of extensions, add-ons, and upgrades; the elimination of bugs and errors; and quality and cost improvements [7].

There has been some research on the opening of software platforms. According to [16], a platform constitutes a "technical architecture that allows compatible complements to use it". Another characteristic of platforms is that they are often centered around a central technology [25]. The platform can be used by other cooperating firms and is usually managed by a platform leader who enables the development of complementary products and services [29,31,36]. Openness here refers to the degree to which the platform does not restrict participation, use and development [11,31]. Opening platforms faces two conflicting goals: adoption and appropriability [40]. On the one side, firms need to appropriate some parts of the economic benefits associated with their platform, however, they also need other firms to adopt their platforms and therefore share some of the economic benefits with their partners. An ES can be conceptualized as a platform because similar to platforms it can be defined as a technical architecture enabling compatible elements to be built on it [16]. However, the existing research on the opening of platforms falls short for ES, as they have unique characteristics not considered in this research. In the next section, ES and their fundamental characteristics will be shown.

3 Embedded Systems

In this chapter, embedded systems and their particular characteristics will be discussed. ES, as it has been mentioned in the introduction, are dedicated computer systems which are embedded in a device to realize a specific function [4,30]. The application areas of ES are very widespread, with applications in the following fields: automotive electronics, aircraft electronics, trains, telecommunication, medical systems, military applications, authentication systems, consumer electronics, fabrication equipment, smart buildings and robotics [28]. In contrast to personal computers, ES are constrained in their hardware and software capabilities [30]. Another important characteristic of

ES is that a user normally cannot change the functionality of the system [20]. In order to provide new services or to change the functionality, the ES must be implemented a way, that new applications do not compromise its dedicated function.
The requirements posed on ES are also quite different compared to normal personal computers: they have to fulfill real-time constraints, both time-sensitive and time-critical constraints [5]. They also have to fulfill higher reliability requirements. Moreover, ES are typically constrained regarding their hardware capacities, for instance regarding processing capabilities, energy consumption, memory and other hardware characteristics [30]. An overview of the characteristics of ES can be seen in Table 1. Due to the high diversity of ES, to be regarded as an embedded system, not all of these characteristics must be present, but a computer system can be classified as an ES, when it fulfills most of these characteristics [28]. Therefore, general-purpose computer systems and embedded system cannot always be clearly differentiated, as the example of smart phones demonstrates. Mobile phones are an example of ES, but microprocessors in smart phones are not dedicated to a specific application anymore. Although ES can be quite different, because of these common characteristics common design approaches are needed [28].

Table 1 Characteristics of Embedded Systems according to [28]

Dependability	Encompasses Reliability, Maintainability, Availability, Safety and Security
Efficiency	Can be measured in energy consumption, run-time efficiency, code size, weight and cost
sensors and actuators	Integrated in the environment through sensors and actuators
Real-time constraints	Computations must be finished in a certain time frame, could be soft or hard real-time constraints
Reactive systems	System execution is shaped by the environment
Hybrid systems	Include analog and digital parts
Dedicated user interface	Realized for instance through push buttons, steering wheels, pedals etc.
Dedicated towards a specific application	Contain specific software which accomplishes a certain task

Traditionally, ES are designed in a closed fashion where the whole software stack is provided by the device manufacturer. Except for firmware upgrades, the software stack does not get altered. Nowadays, due to the increasing complexity and functionality of ES, they are more and more becoming like general-purpose systems [1,21]. For instance, applications originally written for PCs can now be found in smart phones [21]. Therefore, one of the characteristics of ES, namely only being dedicated to a specific application does not apply to every case anymore. This development offers potential for delivering new kinds of innovative functionalities in former closed ES. However, to realize this ES firms also need to open their innovation processes. The opening of innovation processes will be discussed in the next section.

4 Open Innovation

Since Chesbrough has coined the paradigm of open innovation in 2003 [9], there has been extensive research on the opening of the innovation processes to external parties [12]. Open Innovation, in contrast to traditional innovation processes which mainly take place inside the R&D departments of firms, aims at opening the innovation processes to other actors inside and outside the company [9,33].

4.1 The three core open innovation processes

Gassmann and Enkel (2004) categorized open innovation into three core processes: the outside-in process, the inside-out process and the coupled process [15] The outside-in process aims to integrate external actors like suppliers or customers to benefit from external knowledge by increasing the innovativeness. By choosing the inside-out process, companies externalize some of their knowledge in order to commercialize their ideas faster on the market than it would be possible internally. This can for instance be done by licensing intellectual property (IP) and/or providing knowledge to other companies in order to benefit from multiplying technology [15]. The coupled process combines both of these processes (incorporating external knowledge and bringing ideas to the market) by working together with other firms in strategic networks. In these strategic networks, knowledge is created through relationships between specific partners, e.g. in consortia, joint ventures or alliances [15]. The characteristics of firms relying on the three core OI processes according to [15] can be seen in Table 2. These characteristics have been collected by a sample of 124 companies and therefore provide a generalized view on the applicability of the three processes.

Table 2 Characteristics of ES according to Gassmann & Enkel (2004)

Outside-In Process	Inside-Out Process	Coupled Process
Low tech industry for similar technology acquisition	(basic) research-driven company	Standard setting (pre dominant design)
act as knowledge brokers and/or knowledge creators	Objectives like decreasing the fixed costs of R&D, branding, setting standards via spillovers	Increasing returns (e.g. in the mobile industry through multiplying technology)
highly modular products		Alliance with complementary partners
high knowledge intensity		Complementary products with critical interfaces
		Relational view of the firm

Based on the characteristics of these three core innovation processes, in the next part it will be analyzed how they can be implemented in the context of embedded systems. For this purpose, the implications of the characteristics of ES on characteristics of firms relying on the three core OI processes will be shown.

4.2 Implications of the characteristics of ES on the Outside-In Process

Low tech industry for similar technology acquisition

According to [15], firms applying outside-in processes, mostly stem from low tech industries where external partners provide input for developing new technologies. Especially in markets with high competition, firms need to differentiate themselves with innovative functionalities. Firms producing embedded systems could be found both in low tech as in high tech areas. Those ES could be quite primitive regarding their functionality, therefore only low cost hardware would be needed and the software only would have to fulfill simple tasks. But they could also be quite complex as for example in the automotive domain. Therefore, there do not seem to be direct relations between the characteristics of ES and this characteristic of the outside-in process. However, further validation needs to be carried out to provide a comprehensive answer.

Knowledge brokers and/or knowledge creators

In the past, firms deciding on the outside-in process were SMEs which had the role of knowledge creators or brokers to bigger companies [15], however, Gassmann & Enkel [15] state that this refers to past data and company size does not play a big role anymore for firms being knowledge brokers and/or creators. Thus, this characteristic will not be explored further in this paper.

Modularity

According to [2], “modular systems are made up of components that are highly interdependent within sub-blocks, called modules, and largely independent across those sub-blocks” [2]. Complex systems can be subdivided into discrete parts communicating with each other by relying on standardized interfaces as part of a standardized architecture [26]. Due to the independence among different modules, changes in a specific module normally do not influence other modules [2]. Concerning product design, modularity is beneficial when flexibility and rapid innovation are demanded [13,39]. The increase in product innovation is attributed mainly to autonomous and modular innovation [3,13]. ES typically consists of several separate layers [30], thus enabling modularity. Although a layered design provides abstractions from lower levels, applications built on top of the ES architecture must not be allowed to violate real-time constraints and dependability requirements. Modularity is also limited in ES due to its hybrid nature. The physical constraints inherent in ES, software and hardware in ES often needs to be designed simultaneously [14]. Often, modularization is centered around intellectual property (IP). IP-oriented modularization can be used as a tactic to balance value creation and value capture when opening their systems [22]. Therefore, decisions regarding the externalization of IP are mainly dependent on business model decision. According to Henkel & Baldwin (2009) [22], providing open access to some parts of the platform can be the most effective way to increase innovation and value creation in some instances. Decisions on giving up control over intellectual property in ES however, is not only a matter of value creation and value capture, but is also determined by characteristics of ES as well. As cost efficiency is one of the characteristics of ES, externalizing IP for complementary development could also help ES firms to reduce development costs. Another motivation for licensing is the potential reuse of components. However, for safety-critical systems, higher risks are involved, as failures often can be found at interfaces of logically correct components [34].

Knowledge intensity

Firms with high knowledge intensity often tend to outside-in innovation, when the required know-how cannot be acquired inside the firm [15]. Developing for ES requires possessing extensive domain knowledge, therefore, ES developers are usually control engineers and mechanical engineers, which have a thorough understanding of the physical characteristics of the device and the environment where it operates [27]. As ES are hybrid systems, the initial design of ES involves both hardware and software design. The tight coupling of HW and SW in ES requires more know-how than traditional software development [24], increasing with the complexity of the ES. For systems which require a high degree of domain knowledge, open innovation processes thus are confined to experts, especially when it comes to the core functionalities of the system. For the development of additional applications on top of the base system, the complexity involved can be reduced by providing interfaces for external developers. This has for instance taken place in the smartphone domain, where the base system is essentially closed, but interfaces for application development are provided. For devices in which the ES part played only a minor role so far, not so much know-how would be required. But when they plan to implement more innovative functionalities through software, additional know-how would be needed. Sensors and actuators are a domain for which it could be beneficial to acquire external know-how, because writing software for them requires developers to have knowledge about the physical characteristics of the device and its environment which may not be present in the company. It can be seen that implementing the outside-in process is in some aspects restricted by ES characteristics. The results are also depicted in Table 3 later in this paper.

4.3 Implications of the characteristics of ES on the Inside-Out Process

Research-driven companies with objectives like decreasing the fixed costs of R&D, branding, setting standards via spillovers

According to Gassmann & Enkel (2004) [15], companies relying on the inside-out process are mostly research-driven companies with broad application fields which aim at reducing the fixed costs of R&D and mitigating risks by sharing them with partners. However, due to the dedication to a specific function in many ES, embedded systems are often seen in terms of their cost efficiency and not as a source of innovation. Therefore, most ES do not have broad application fields. However, with the tendency of ES to become cheaper and at the same time having more and more performance, the demand for innovative functions is increasing [8]. Furthermore, ES firms are not necessarily research-driven companies, as for example consumer goods manufacturers often compete more on prices than on new technologies. However, in domains such as the automotive domain, more and more innovative functions are implemented via software [10]. Furthermore the increasing performance of microchips combined with decreasing costs lead to a higher demand of innovative functions by the customers. Additionally, in highly competitive markets, focusing more on innovative functions could help firms to differentiate themselves. For branding, when firms have internal capabilities for the development and commercialization of products but do not possess a brand on a specific market, we did not find implications of ES characteristics. The goal of setting standards will be analyzed separately in the next section as it is also a characteristic determining the coupled process.

Standard setting

Standards for embedded systems can be divided into market-specific standards and general-purpose standards and standards which apply both of these two cases. Market-specific standards refer to similar types of embedded systems, for example according to technical or end-user characteristics with [30]. Such standards can for instance be defined by industry consortiums such as AUTOSAR in the automotive industry. One of the goals of AUTOSAR is to enable the interoperability among IP (software, hardware and tools) [35]. General-purpose standards are not limited to a specific class of embedded devices, but could be adopted in other ES and non-ES, for instance programming-language based standards [30]. For ES firms, relying on standards facilitates involving external actors for open innovation processes. For instance, the Java language is such a standard which works with a high variety of different hardware architectures [30]. Therefore, efficiency as a property of ES could be increased by the implementation of standards in the design of ES. In terms of opening embedded systems, the question of standards is crucial in order to ensure interoperability. It is of particular relevance in complex ES like in the automotive domain where many suppliers and partners work together in order to provide an integrated solution. Therefore, for open innovation in the business-to-business sector, the implementation of market-specific standards in addition to general purpose-standards needs to be ensured. However, due to the tight coupling of embedded software to the hardware, standardization is often only possible to a certain degree [41].

Based on this analysis, the characteristics of ES provide only minor challenges to the Inside-Out process, however, the setting of standards of ES firms can be seen as a requirement for external participation.

4.4 Implications of the characteristics of ES on the Coupled Process

Standard Setting

The characteristic of standard setting has already been analyzed for the Inside-Out process and applies to the coupled process as well.

Increasing returns by multiplying technology

Increasing returns can be exploited by firms through multiplying their technology by setting industry standards, as it has for instance been taken place in the mobile industry with the MMS or the UMTS standard or the polyphone ring tunes [15]. In order to establish those standards, industry-wide strategic alliances are required. This strategy is of particular relevance in the case of network effects where the value for customer increases when more participants join the network [32]. With the increasing connectivity of ES, telecommunication producers could play a key role [41] in providing solutions. Implications of the characteristics of ES on how firms can profit by multiplying technology were not found in this paper.

Alliances with complementary partners

Alliances with complementary partners might be in some cases a more promising approach for ES producers than a broader opening their system to external partners. Especially, when the integration of components developed by other parties is subject to high complexity, having strong ties to these partners would be beneficial to manage the integration process. For instance, in the case of automotive software, the integration of components of safety-critical components is a major challenge [35]. Due to the hybrid nature of ES, software engineering and mechanical and electrical engineering are part of overall system engineering, which makes coordination more challenging [17]. Therefore, with increasing complexity of the system, alliances would be preferable to loose cooperation. Regarding the different layers of the embedded system architecture, the more critical parts of the systems are affected (in terms of dependability and real-time requirements), the more ES firms should seek closer alliances. According to a study of the significance of the ES sector in Germany from Bitkom (2008) [41], many ES firms see potentials for synergy among firms from different industries which face similar challenges.

Complementarity of products

Drawing from the research on platforms, complementarity is an important design goal in two-sided markets, with the platform owners differing from the application developers [19]. One of the motivations for opening technologies is to stimulate the development of complementary products. For instance, revealing source code is a means to increase complementarity [23]. However, simple ES architectures often do not provide software layers abstracting from the hardware layers to enable application development [30]. By providing additional software layers, for instance an operating system layer, developing applications is facilitated. By providing interfaces for developers (Application Programming Interfaces), it is easier for application developers to develop complementary applications. However, those interfaces are often not offered by the manufacturers of embedded devices due to economic reasons and technical challenges. For platform vendors, it is often more profitable to provide their own applications. Furthermore, they often do not want to lose control over their platform [18]. The opening of ES in the form of providing interfaces bears risks as well, especially security risks, e.g. viruses and worms [21]. Besides these security threats, safety issues also prevent firms from allowing complementary SW development. For example, in the automotive industry, when software stems from different suppliers, the integration of safety-critical sub-components requires strong methodology and discipline to control the compliance to this methodology of partners and suppliers [35].

Relational view of the firm

The characteristic "Relational View of the firm" denotes a cultural aspect of firms, namely the ability to sustain "the right balance of give and take" which is required when working in strategic alliances and joint ventures [15]. We did not find direct influences of the characteristics of ES on this aspect.
Similar to the Outside-In process, the characteristics of ES are crucial to the successful implementation of the Inside-Out process. Of particular importance in this context is to ensure tight coordination between the firms and external partners.

4.5 Results

The results of this analysis have been depicted in Table 3. It shows how the different characteristics of ES affect the three core OI processes. We did not find implications of the ES characteristics on every OI process, so some of the cells are empty. However, these results still need to be empirically validated.

Table 3 Implications of the characteristics of ES on the three core OI processes

	Outside-In Process	**Inside-Out Process**	**Coupled Process**
Dependability	Dependability more difficult to ensure with high modularity; Safety requirements limit licensing possibilities		Tight coordination among partners or in alliances required because of dependability; Safety requirements limit potential complementarity
Efficiency	Aim of cost efficiency could better be attained by outside licensing	Required efficiency drives implementation and development of standards	Required efficiency drives implementation and development of standards
Sensors and Actuators	External know-how could be beneficial for sensors and actuators due to high knowledge intensity		
Real-time constraints	Design for modularity needs to ensure real-time constraints, e.g. by tight coordination with partners		

Reactive systems			
Hybrid Systems	Higher knowledge intensity due to dichotomy of HW and SW, which could be met by external know how; Physical constraints hinder separate, modular design	Tight coupling of HW and SW complicates standard implementation	Tight coupling of HW and SW complicates standard implementation; Hybrid aspects add to complexity and therefore needs strong coordination
Dedicated user interface			
Dedicated towards a specific application			Long-term trend in ES design could be towards multiple applications and complementarity

The results show, that especially the outside-in and the coupled process are affected by the characteristics of embedded systems, whereas the inside-out process seems to be more independent from ES characteristics. A factor that requires more exploration is the relation between some of the firm characteristics implementing these OI processes. For instance, setting standards can be seen as a facilitator of modularity and complementarity, especially when other parties are involved. Furthermore, as it has already been mentioned, not all kinds of ES have to fulfill these characteristics to the same degree, therefore, some aspects could be more or less relevant when considering a specific type of ES. Regarding the outside-in process, there are some characteristics of ES posing challenges for the involvement of external actors. Especially dependability requirements, real-time constraints and the hybrid composition of ES require tight cooperation between the involved parties. In cases, where these characteristics are not as critical, looser forms of coordination would be imaginable. One aspect of ES presented in all of the three OI processes is the aim to achieve higher efficiency, especially in terms of costs. However, as efficiency in ES development has traditionally been seen as equipping a device with cheap hardware with limited capabilities, there was scarcely potential for innovative applications. By relying on open innovation on the software side, additional efficiency can be gained, even though more hardware resources would be required.

However, these results still have to be empirically validated. But this classification serves as a first understanding about the influence of ES characteristics on open innovation processes. So it is a framework to systematically discuss relevant issues in opening up ES.

5 Conclusion

As research concerning the combination of open innovation with a technical perspective outside the open source development is still scarce, this paper contributes to understand the applicability of open innovation in technical settings, in this case in the field of embedded systems. Based on the characteristics of firms implementing the three core innovation processes according to Gassmann & Enkel (2004) [15], we analyzed the implications of the characteristics of embedded systems on the applicability of open innovation in this field. As a result, it came out, how these characteristic either facilitate or hinder the three open innovation processes. Of these three innovation processes, each of them could be used in the context of ES, however especially when relying on the outside-in process and the coupled process. Ensuring the requirements of ES poses some challenges. As this analysis was based on the characteristics of firms implementing the three core OI processes, it provides guidance for ES firms in the implementation of OI processes. Further research should focus on validating the proposed framework and exploring missing factors of ES influencing OI as well as identifying missing characteristics of ES firms which determine the applicability of OI. As embedded systems are quite diverse, the evaluation should incorporate different classes of embedded systems to provide a comprehensive picture.

References

[1] Aguiar, A. and Hessel, F. Embedded systems' virtualization: The next challenge? Rapid System Prototyping (RSP), 2010, (2010).

[2] Baldwin, C. and Henkel, J. The impact of modularity on intellectual property and value appropriation. Working Paper, (2011).

[3] Baldwin, C.Y. and Clark, K.B. Design Rules: The Power of Modularity. Blackwell, Malden, MA, 2000.

[4] Barr, M. and Massa, A.J. Programming Embedded Systems: With C And Gnu Development Tools. O'Reilly Media, Inc., 2006.

[5] Berger, A. Embedded Systems Design: An Introduction to Processes, Tools, and Techniques. CMP Books, Manhassett, NY, 2002.

[6] Boudreau, K. Does Opening a Platform Stimulate Innovation? Effects on Modular and Systemic Innovation. MIT Sloan Research Paper 4611-06, (2006).

[7] Boudreau, K. Open Platform Strategies and Innovation: Granting Access vs. Devolving Control. Management Science 56, 10 (2010), 1849-1872.

[8] Broy, M. Challenges in automotive software engineering. Proceeding of the 28th international conference on Software engineering - ICSE '06, (2006), 33.

[9] Chesbrough, H. Open innovation: The new imperative for creating and profiting from technology. Harvard Business School Press, Boston, MA, 2003.

[10] Ebert, C. and Jones, C. Embedded software: Facts, figures, and future. Computer 42, 4 (2009), 42–52.

[11] Eisenmann, T.R., Parker, G., Alstyne, M.V., and Eisenmann, T. Opening Platforms : How , When and Why ? Harvard Business School Working Paper 09-030, (2008).

[12] Elmquist, M., Fredberg, T., and Ollila, S. Exploring the field of open innovation. European Journal of Innovation Management 12, 3 (2009), 326-345.

[13] Ethiraj, S.K. and Levinthal, D. Modularity and innovation in complex systems. Management Science, September (2004), 159–173.

[14] Fernandes, J.M., Lamb, L.C., Machado, R.J., and Wagner, F.R. Recent advances in model-based methodologies for pervasive and embedded software. ACM SIGSOFT Software Engineering Notes 34, 5 (2009), 37.

[15] Gassmann, O. and Enkel, E. Towards a theory of open innovation: three core process archetypes. R&D management conference, (2004).

[16] Gawer, A. Platform dynamics and strategies: from products to services. Platforms, Markets and Innovation,, (2009).

[17] Graaf, B., Lormans, M., and Toetenel, H. Embedded software engineering: The state of the practice. Software, IEEE 20, 6 (2003), 61–69.

[18] Gunter, C. Open APIs for embedded security. Proceedings of the European Conference on Object Oriented Programming, (2003), 225-247.

[19] Haruvy, E., Sethi, S.P., and Zhou, J. Open Source Development with a Commercial Complementary Product or Service. Production and Operations Management 17, 1 (2008), 29-43.

[20] Heath, S. Embedded systems design. Butterworth- Heinemann, 2003.

[21] Heiser, G. The role of virtualization in embedded systems. Proceedings of the 1st workshop on Isolation and integration in embedded systems - IIES ’08, (2008), 11-16.

[22] Henkel, J. and Baldwin, C. Modularity for value appropriation: Drawing the boundaries of intellectual property. Working Paper, 2009.

[23] Henkel, J. Selective revealing in open innovation processes: The case of embedded Linux. Research Policy 35, 7 (2006), 953-969.

[24] Henzinger, T. a. and Sifakis, J. The Discipline of Embedded Systems Design. Computer 40, 10 (2007), 32-40.

[25] Iansiti, M. and Levien, R. The keystone advantage: what the new dynamics of business ecosystems mean for strategy, innovation, and sustainability. Harvard Business School Press, (2004).

[26] Langlois, R.N. Modularity in technology and organization. Journal of Economic Behavior & Organization 49, 1 (2002), 19-37.

[27] Liggesmeyer, P. and Trapp, M. Trends in embedded software engineering. Software, IEEE 26, 3 (2009), 19–25.

[28] Marwedel, P. Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems. Springer, 2010.
[29] Mikkola, J.H. and Skjøtt-Larsen, T. Platform management: Implication for new product development and supply chain management. European Business Review 18, 3 (2006), 214-230.
[30] Noergaard, T. Embedded Systems Architecture: A Comprehensive Guide For Engineers And Programmers. Elsevier, 2005.
[31] Parker, G.G. and Van Alstyne, M.W. Two-Sided Network Effects: A Theory of Information Product Design. Management Science 51, 10 (2005), 1494-1504.
[32] Parker, G.G. and Van Alstyne, M.W. Two-Sided Network Effects: A Theory of Information Product Design. Management Science 51, 10 (2005), 1494-1504.
[33] Reichwald, R. and Piller, F. Interaktive Wertschöpfung: Open innovation, Individualisierung und neue Formen der Arbeitsteilung. Harvard Business School Press, Boston, 2009.
[34] Saglietti, F. Licensing Reliable Embedded Software for Safety-Critical Applications. Real-Time Systems 28, 2/3 (2004), 217-236.
[35] Sangiovanni-Vincentelli, A. and Di Natale, M. Embedded System Design for Automotive Applications. Computer 40, 10 (2007), 42-51.
[36] Schlagwein, D., Schoder, D., and Fischbach, K. Openness in the Orchestration of Ecosystems : A Resource-based Perspective. Working Paper, 2010 (2010).
[37] Shapiro, C. and Varian, H.R. Information Rules: A Strategic Guide to the Network Economy. Harvard Business School Press, Boston, MA, 1999.
[38] Turley, J. Embedded Systems-Connecting Everyone to Everything. Technology Radar Feature Paper, (2009).
[39] Ulrich, K. and Eppinger, S. Product design and development. McGraw-Hill, New York, 2000.
[40] West, J. How open is open enough? Melding proprietary and open source platform strategies. Research Policy 32, 7 (2003), 1259-1285.
[41] Ziegler, S. and Müller, A. Studie zur Bedeutung des Sektors Embedded-Systeme in Deutschland. BITKOM, Berlin, Germany, 2008.