

THE COMING OF LIGHTWEIGHT IT

Bendik Bygstad

Dept. of Informatics, University of Oslo, Norway, bendikby@ifi.uio.no

Abstract

Two current trends are changing the IT industry and the ways we develop IT solutions; we suggest to call them heavyweight and lightweight IT. Heavyweight IT is the traditional systems and databases, which are becoming more sophisticated and expensive through advanced integration. Lightweight IT is the new paradigm of mobile apps, sensors and bring-your-own-device, also called consumerisation or Internet-of-Things. The key aspect of lightweight IT is not only the cheap and available technology as such, but the fact that its deployment is frequently done by users or vendors, bypassing the IT departments. Drawing on the theoretical lens of generativity, our research questions are, how is generativity different in heavyweight and lightweight IT, and what is the generative relationship between heavyweight and lightweight IT?

These questions were investigated through a study of four cases in the health sector. Our findings show that (i) generativity unfolds differently in heavyweight and lightweight IT and (ii) generativity in digital infrastructures is supported by the interaction of loosely coupled heavyweight and lightweight IT. The practical design implication is that heavyweight and lightweight IT should be only loosely integrated, both in terms of technology, standardisation and organisation.

Keywords: Lightweight IT, heavyweight IT, digital infrastructures, generativity

1 Introduction

In June 2014 Apple announced a partnership with EPIC, the dominant vendor of Electronic Patient Journal systems in the USA. Entering the e-health market Apple disclosed two coming products; a Health app that will serve as a personal dashboard for your health and fitness data, and a development platform for 3rd party vendors called HealthKit. The partnership illustrates that two current trends are changing the IT industry and the way we use IT. First, there is an on-going effort to integrate IT silo systems into seamless solutions, by various technologies such as service-oriented architecture and cloud computing. We should regard this as a new wave in software development: the technical and management challenges are significant, and the costs are very high (Sommerville et al., 2012). The solutions are quite advanced, but also more complex. Typical examples are found in health care, finance and the public sector, which all have a legacy of silo systems (Bannister, 2001; Bouwman et al., 2011).

Second, the increasing use of privately owned units, such as smartphones and tablets, in work life has challenged hegemony of the IT departments. “Bring your own device” has quickly become an accepted fact of corporate life; in an Accenture survey 45% of the respondents agreed that “the hardware devices and software applications that I personally use are more useful than the ones provided by work” (Harris et al., 2012). The trend has been named *consumerisation*, and presents organisations with a whole set of new challenges concerning use, security and IT governance (Niehaves et al., 2012).

Interestingly, both trends are responses to the increasing complexity of digital infrastructures, at various levels. The integration efforts are done in order to reduce organisational complexity stemming from silo systems, for example when process change and innovation are hindered by non-integrated solutions (Ross et al., 2006). The consumerisation trend is to a large degree triggered as a response to the increasingly bureaucratic solutions and security mechanisms of corporate IT (Harris et al., 2012).

In this paper we build on the insights from the consumerisation literature and the “Internet-of-Things” research (Etzori et al., 2010), but our approach extends these concepts in arguing that we deal with not only new technologies, but with a new socio-technical knowledge regime. The term knowledge regime has usually been used in sociology and political science to denote a set of actors and organisations that produce and disseminate ideas that affect policy-making (Campbell and Pedersen, 2007). However, some researchers in the IS field, such as Howard-Grenville and Carlyle (2006) have extended and sharpened the concept to deal with organisational processes and technology, and defined it as “the nested connections between the material realities engaged by work practices, the work practices themselves, and the larger collective conventions that reflect and account for the appropriate use of such practices” (p.474).

Central to the rise of the new knowledge regime is that IT-based innovation increasingly is being conducted by non-IT professionals, by deploying cheap and easy-to-use IT. We suggest calling the phenomenon lightweight IT¹, because it is “light” in several aspects: It is typically cheap and easy to use technology, it can often be deployed without IT specialists and it tends to be mobile technology. We believe it is fruitful to regard the two trends as a paradigmatic shift to two different knowledge regimes: *heavyweight and lightweight IT*. Heavyweight IT is becoming increasingly complex and specialized, while lightweight IT emerges as a new innovation arena, allowing non-specialist to experiment with cheap technology. This is also acknowledged in Gartner’s (2014) concept of “bimodal IT”, suggesting two different IT departments: one for traditional IT, focused on stability and efficiency, and one experimental and agile, focused on time-to-market and tight co-operation with business units. Lightweight IT extends this perspective, and we define it as *a socio-technical knowledge regime driven by competent users’ need for IT services, enabled by the consumerisation of digital technologies*.

To study these phenomena we chose the e-health sector, which is currently experiencing strong growth and intensive IT innovation in many rich countries (Christensen, 2011). At the same time it is also characterized by serious challenges: a long history of “IT silo systems” (Bannister, 2001) has created a maze of clinical and administrative information, which has become a serious hindrance for patient-centred health care in most developed economies (EU Commission, 2011). Mitigating these problems has led to national mega-programs, which are expensive and high-risk initiatives (Sauer and Willcocks, 2007, Greenhalg et al., 2010).

While these programmes are dominated by heavyweight IT thinking, a wave of lightweight innovation is entering the e-health field: sensors, apps and tablets are becoming available in the health care area. These solutions are quite diverse; some of them support work processes for health care personnel, and some provide new services, but overall they are not part of the heavyweight architecture (Miorandi et al., 2012). Moreover, these technologies are quite attractive for both clinicians and patients, and vendors market their products and solutions directly. At the moment it is an open question how heavy-

¹ The notions of lightness and heaviness have a long history in European philosophy and literature. In *On The Heavens* Aristotle ascribed absolute weight to the earth and absolute lightness to fire, while the weight of other elements was relative. Parmenides argued that lightness was positive and to be desired, while weight was negative. In *The Unbearable Lightness of Being* Milan Kundera played with the dichotomies of lightness (momentary pleasures) and heaviness (Nietzsche’s idea of “eternal return”, where time is circular). In software engineering the term lightweight methods, such as XP was introduced around 2000, as a contrast to heavyweight methods, such as RUP.

weight and lightweight IT can be integrated, technically and organisationally. In order to deal with the practical issues of integration we need to understand the more fundamental issues of the specifics of heavyweight and lightweight IT, and their interactions. Two research questions are investigated in this paper:

- How is generativity different in heavyweight and lightweight IT?
- What is the generative relationship between heavyweight and lightweight IT?

We proceed by defining heavyweight and lightweight IT. We frame our analysis within the digital infrastructure tradition (Hanseth and Lyytinen, 2010), which is primarily interested in the dynamics of large networks rather than stand-alone applications. To develop our argument we draw on the literature on generativity, i.e. the ability of technical and social elements to interact and recombine to produce new solutions. Four cases within e-health are used to explore and illustrate the research. The key argument is that generativity in digital infrastructures is supported by the interaction of loosely coupled heavyweight and lightweight IT.

2 Lightweight and Heavyweight IT

What is here called heavyweight IT is the mainstream IT as currently delivered by IT departments over the world: back-end solutions such as ERP and other transaction systems, based on databases servers and integration software, such as bus architectures (Rosen et al., 2008). Many organisations within finance, health and public sector are engaged in major efforts to deal with the *IT silo problem*, the fact that a large number of poorly integrated legacy systems constitute a barrier to organisational change and innovation (Bannister, 2001; Bouwman et al., 2011). However, complexity is increasing as systems become more integrated, and the requirements to security and resilience are also increasing. Sommerville et al. (2012) called these interconnected solutions *coalition of systems*, i.e. systems that continually communicate during operation. The more interconnected, the higher risk for unexpected incidents. Therefore there are limits to integration.

Lightweight IT may be seen as complementary to heavyweight; it is well suited for the tasks that heavyweight IT has often failed to support, i.e. the simple and immediate needs of a user. Lightweight IT typically supports work processes with simple applications on cheap technology (Alemdar and Ersoy, 2010). The most obvious example is the “*app*” revolution where small programs on handheld devices substitute heavyweight GUIs, for instance to get information on when the next bus is arriving, or allows you to change your flight schedule while sitting in a meeting. Another example is the use of commercially available sensors in health care, which can easily be connected to mobile devices via Internet or operator’ services.

The IT health sector is currently in the middle of an IT revolution (Christensen et al., 2011). In the heavyweight segment the many clinical silo systems (such as electronic journal, imaging, lab and curve systems) are being connected through various integration solutions, such as service-oriented architecture (Rosen et al., 2008), or replaced by suite systems, such as EPIC (McCarthy et al., 2009). These efforts are using advanced solutions with specialized IT staff. They are also very costly, but promise to solve key parts of the silo problem of the past.

The lightweight segment is a thriving arena of intense innovation, ranging from advanced bodily sensors and surgery robots to Internet tablets with Facebook for the elderly (Sherer, 2014). It still makes sense to characterize these solutions as lightweight, because they are usually initiated by medical, provided by actors outside heavyweight IT, and strongly connected to work processes. Lightweight solutions are often patient-centric, but this is not a defining attribute, since also heavyweight solutions can be patient-centric.

It is notable the development cultures differ substantially; heavyweight IT is increasingly being professionalized while lightweight IT is becoming a large innovation arena. The explosive growth of 3rd

party app developers (Bergvall-Kåreborn and Howcroft, 2011), illustrates this trend. Also, the IT architecture is quite different; heavyweight IT is based on highly structured solutions designed by enterprise architects, while lightweight IT often has emerged as what DeLanda (1998) has called *meshworks*, i.e. heterogeneous networks of technological agents and users. The heavyweight development culture can be described as the systematics and quality oriented ethics of software engineering, while lightweight culture is more experimental and innovation oriented.

	Heavyweight IT	Lightweight IT
Profile	Back-end: Supporting documentation of work	Front-end: Supporting work processes
Systems	Transaction systems	Process support, apps, BI
Technology	Servers, databases, enterprise bus technology	Tablets, electronic whiteboards, mobile phones
IT architecture	Centralized or distributed	Meshworks
Owner	IT department	Users and vendors
Development culture	Systematics, quality, security	Innovation, experimentation
Problems	Increasing complexity, rising costs	Isolated gadgets, security
Discourse	Software engineering	Business innovation

Table 1. Heavyweight and lightweight IT

Although, as table 1 illustrates, the two domains are moving in opposite directions, they are also complementary. Lightweight IT is to a large degree dependent on heavyweight IT as a platform and as a data repository. The reverse is less obvious, but still true: heavyweight IT is dependent on lightweight IT for innovation and organisational agility.

A salient example of connected heavyweight and lightweight IT is seen in the so-called *platform systems*. A platform is the enabling centre of ecosystems such as Google, Amazon and Apple, and allows them to grow through the efforts of others. Platform architectures are “modularisations of complex systems in which certain components (the platform itself) remain stable, while others (the complements) are encouraged to vary in cross-section or over time”(Baldwin and Goddard, 2008).

The generativity of platforms arises from the fact that the owners of the platform realize that they do not necessarily have the most creative ideas on how to use it, and invite others to innovate. Yoo (2013) noted that platform owners need to maintain a delicate balance of control; if the control of third-party contributors becomes too tight the ecosystem loses its ability to generate innovation. On the other hand, if the platform is too malleable, it easily becomes too fragmented to serve as a platform. Platforms are one way to connect heavyweight and lightweight IT, but not the only one. However, in order to analyse this relationship in more depth, we draw on the concept of generativity.

3 Theoretical Lens: Generativity

The term generativity has old philosophical roots, going back to Leibniz (Smith, 2011), but is commonly used in modern sciences such as evolutionary biology, cybernetics and linguistics to express the basic idea that the observed complexity of a phenomenon (such as biological diversity, social systems and language) can be traced back to some basic elements and their mechanisms for interaction. (Phelan, 2003).

Jonathan Zittrain (2006) introduced the term *generative technology*, defined as “a technology’s overall capacity to produce unprompted change driven by large, varied, and uncoordinated audiences (p.1980)”. Zittrain defined generativity in more detail as a function of a technology’s capacity for leverage across a range of tasks, adaptability to a range of different tasks, ease of mastery, and accessibility. An excellent example of a generative technology is the TCP/IP and http protocols of the Internet.

Generativity, however, is a socio-technical concept, including users and developers involved. David Lane (2011) argued that a community's capacity to innovate is determined by its *generative relationships*, i.e. the on-going discourse between the actors on their interpretation and use of artifacts. Likewise, Avital and Te'eni (2008) found that the extent to which innovation will take place also depends on an appropriate combination of a generative technology and a generative collective of users and developers. DeLanda, (2006) in a similar way, defined generativity as an entity's "capacity to interact" with other entities.

This calls for a *relational* approach to the object of study; it is neither the specific attributes of the technology nor the attributes of people and organisations we wish to understand, but the emerging relationships between them. One way to frame this is the theory of *affordances and constraints*; affordances express the potential action for humans with a certain technology, while constraints refers to ways that technology sets limitations for action (Majchrzak and Markus, 2012). For example, an affordance between an able boy and a bicycle is called "cycling". An affordance between a patient, a sensor, an iPad and nurse could be expressed as "continuous pulse surveillance". An affordance exists whether it is actualized or not, as a possibility for goal-directed action (Volkoff and Strong, 2013).

Digital technology tends to offer a richer set of affordances than physical products (Yoo, 2013). For example, a bicycle is excellent for bicycling, but cannot be used for much else. In contrast, the past decade has showed that an able developer with a smart phone can identify an almost endless array of new affordances for various groups and purposes. It is clear from this that generativity is an *emergent* phenomenon (Elder-Vass, 2007); it is not an attribute of an object (such as technology), but rather the potential outcome of the interaction of different entities. That is, the outcome is dependent on, but not reducible to the entities. In our context we take generativity to mean the outcome of the interaction between knowledgeable people and flexible information technologies. Many researchers, such as Yoo et al. (2012) and Leonardi (2011) have analysed the huge innovation potential that lies in the creative recombination of technical and social elements of digital infrastructures.

In the context of digital infrastructure generativity is a key attribute, because infrastructures evolve through innovation and organic growth, rather than by management interventions (Hanseth and Lyytinen, 2010). In understanding these dynamics, we build on the contribution of Henfridsson and Bygstad (2013) that describe the evolution of digital infrastructures as the interplay between three self-reinforcing generative mechanisms (see figure 1):

- *Innovation*: The creative combination of social and technical elements in order to create new services
- *Adoption*: The recruitment of users through easy-to-use solutions, which allows more investments
- *Scaling*: The expansion of the network to include more partners to provide more services.

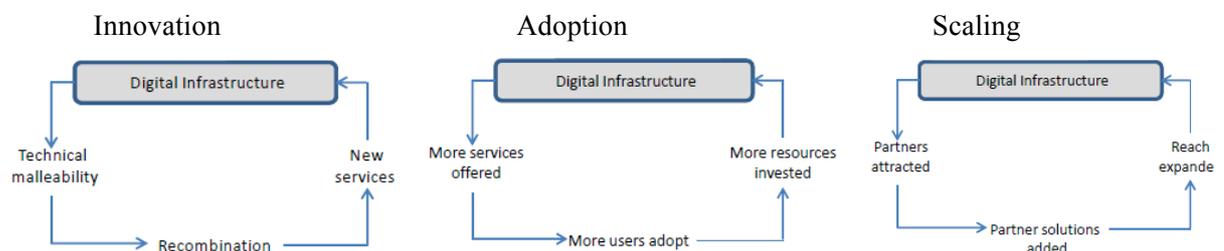


Figure 1. Generative mechanisms

A successful digital infrastructure is self-reinforcing, i.e. it recursively feeds on itself. The mechanism in figure 1 is self-reinforcing: the technical malleability of infrastructure creates a space of possibility, which can be used to recombine components into new services. These services increase the space of

possible re-combinations, and so on. The three mechanisms may also reinforce each other: Adoption lead to more resources that can be used for more innovation, while scaling provides more partners with more services, which leads to more adoption. It is documented over a broad range of different types of digital infrastructures that the interaction of the three mechanisms explains successful cases (Henfridsson and Bygstad, 2013). We will build our analysis on this framework.

4 Method

Investigating these issues empirically is demanding, because generativity is a phenomenon that is not easily observed. We need to be specific regarding the context we are addressing, because generativity may mean quite different things depending on context, depending on the inner workings of generative processes. In order to frame the investigation we make two methodological assumptions on generativity in the context of heavyweight and lightweight IT.

First, generativity should be interpreted as a characteristic of a digital infrastructure, not as an attribute of isolated solutions. Digital infrastructures are assemblages of interconnected systems, including technology, users and developers (Hanseth and Lyytinen, 2010). Thus, generativity is a systemic phenomenon, and we should study it accordingly. Second, it should be observed as a process, unfolding over time. Digital infrastructures are not specified and designed, but evolve through innovation, adoption and scaling.

4.1 Research approach

The study was part of a larger research programme within e-health. Two parallel case studies were conducted; an e-health mega-project in the South-Eastern Norway Regional Health Authority (“Regional Authority”), with predominantly heavyweight IT, and a study of welfare technologies in Norway and Japan, with predominantly lightweight IT.

From these we selected four cases (table 2) in order to explore and illustrate generativity. The cases were selected because they offered an opportunity to investigate generativity in the context of heavyweight and lightweight IT. They should be regarded as relatively *typical* cases (Gerring, 2007); they were all successful, but not spectacular examples. They were also small enough to warrant a relatively detailed analysis of generativity, but they were also parts of larger initiatives and infrastructures.

4.2 Data Collection

Data was collected with a clear focus on identifying and understanding the generative mechanisms of each of the digital infrastructures (Henfridsson and Bygstad, 2013):

- To study *innovation* we collected information on the actors and events, and the recombination of technical and social elements. We documented in detail how a new service was conceptualised, who was involved, and how the development unfolded.
- For charting *adoption* we researched how the service was marketed and the recruitment of users. We also observed practical use, and to the extent the growth of users generated more resources.
- To document *scaling* we collected data on the technical and social openness of the solution, and collected information on active partners. In particular we documented the practicalities of linking into the infrastructure for external partners.

Case	Type	Owner/ Location	Data collection
The Integration Factory	Heavyweight	Regional Authority, Oslo	Longitudinal study, with 25 interviews, document analysis and observation
Kasama City	Heavyweight	Kasama City, Japan	Full-day session with managers and developers, and technical documentation and demo.
St.Hanshaugen	Lightweight	Oslo Municipality	Interviews with managers, vendor and users, and observing use
Køge Hospital	Lightweight	Køge, Denmark	Interviews with managers and users, and observing use. (The case is also documented in Hertzum and Simonsen, 2010).

Table 2: Cases

Data Analysis

Building on a critical realist approach (Elder-Vass, 2007; Wynn and Williams, 2012), our data analysis builds on the context-mechanism-outcome (CMO) devised by Pawson and Tilley (2009). It allows for the analysis of possible configurations of mechanisms and relevant context-variation to explain a particular outcome illustrated in figure 2.



Figure 2. The CMO scheme

We analysed each case as follows: The context was the organisational and technical setting, with heavyweight or lightweight IT. The mechanisms were the three basic generative mechanisms of infrastructure growth; innovation, adoption and scaling. In analysing the differences between heavyweight and lightweight IT, we compared and contrasted how innovation, adoption and scaling processes unfolded, in the two different knowledge regimes. The outcome was the result of the process, described in terms of degree of success for the stakeholders.

For example, in the first case, the Innovation Factory, we analysed each generative mechanism: *how* innovation of new services unfolded, *how* user adoption in the hospitals happened, and *how* the solution scaled as new actors were included into it. Then we analysed the impact of context, and found that heavyweight IT had a strong influence on the processes: that the innovation process included a highly specialised IT environment, that adoption was mandatory and supported by standardised procedures, and that new partners were included by specifications and contracts. Finally, we noted that the outcome was perceived as positive by stakeholders. The result of the data analysis is illustrated in table 3.

Cases

Case 1: The Integration Factory

Context: In 2012 the Regional Authority (which manages 13 hospitals, including largest hospital in Norway, Oslo University Hospital), started a large IT initiative, in order to consolidate and integrate their most important silo systems. These included electronic patient systems, lab and imaging systems, i.e. the core systems for clinical support. To integrate the various systems, a regional integration platform was introduced, building on an enterprise service bus architecture, shown in figure 3. The Biz-Talk solution enables routing and transformation of messages between DIPS and the other systems (“Fagsystem”) within the Oslo University Hospital, and with systems in other hospitals (via the Gateway) and other registers.

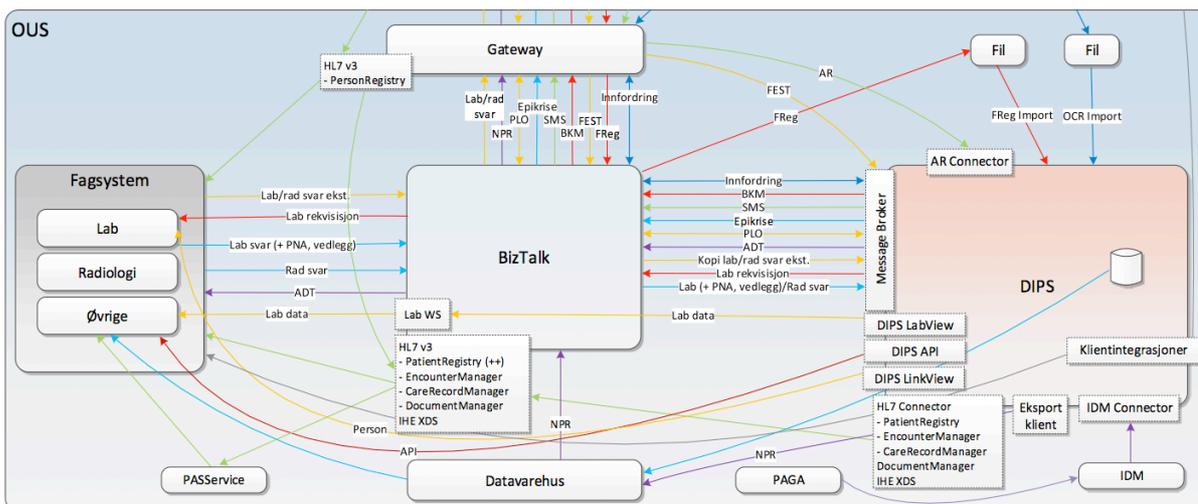


Figure 3: The Integration Platform at Oslo University Hospital (source: Digital Renewal Programme)

Generative mechanisms: Innovation of new services (for example allowing a doctor to examine the x-ray images of a patient from another city, or allowing patients to inspect their treatment history) is dependent on being able to recombine elements from different systems. This makes integration of heavyweight IT a continuous effort. Acknowledging this, the Regional IT department established an organisational unit called the *Integration Factory*, with more than 25 specialised developers. The Integration Factory is a specialised programming unit for dealing with the BizTalk solutions; it receives order from the on-going projects, and programmes the formats and routing of messages and web services. The solution scales well, since it can integrate new systems in the same way.

Outcome: The Integration Factory works as an exchange of data elements, such as messages and service calls. Commented one of the managers of the unit:

“We established the Integration Factory in one of the smaller organisational units, in order to connect the various systems. The regional integration platform breaks up the silo systems by allowing a doctor in one hospital to access not only patient data from another hospital, but also his lab results and imaging files. In establishing the Factory the we focused on two crucial issues; we chose one technology that could provide this functionality, and we spent time and resources to build a specialized and highly competent team of developers”.

In other words, it affords the specialized IT professionals the ability to recombine information from different sources into more integrated services to medical personnel.

Case 2: St.Hanshaugen

Context: This is a town district in Oslo, with around 25.000 inhabitants. The Home-based Care Section has organised a lightweight IT solution for connecting the elderly with the municipal services.

Generative mechanisms: The Norwegian vendor Dignio developed the solution, in close co-operation with the managers of the town district. It provides the employee (who might for instance, be a nurse) with a tablet. Adoption was easy, as the tablet may have a list of (f.ex.) the 30 home residents that is today's responsibility for the nurse. Each elderly is linked to the tablet, in different ways: one may suffer from dementia, and “his” line in the tablet will show a sign if he has forgotten to lock the door at night. Another may suffer from Chronic Obstructive Lung Disease, and will update (on her own tablet) the status of her health and activities every day, which will be read by the nurse. A third may suffer from a heart disease, and has a sensor from the hospital, which may trigger an alarm at the nurse's tablet. The solution is rather flexible, since the vendor can connect various devices (with various transmission solutions and standards) to the nurse's tablet. It will be possible to relay messages to family members.

Outcome: As Deputy Director Sven Bue Berger enthusiastically told us, “Everyone is happy: the elderly feel safer and better monitored than before - without getting the doors overrun by unnecessary visits from care personnel. My staff has more control and an easier work day, and we avoid unnecessary and expensive call-outs”.

Case 3: Køge Hospital, Denmark

Context: Køge is a town south of Copenhagen. Our case was the emergency department, which deals with acute illness; patients may arrive by ambulance or just turn up, often driven by family members. A key challenge earlier was poor logistics; registering the patients, deciding priorities, communicating waiting time, allocating the right doctor or nurse to the right patient, ordering lab tests and x-rays, conducting patient surveillance during the first critical hours. In addition, family should be informed, and other hospital departments (or other hospitals) should be informed of transfers. All these activities are important, and in a hectic work situation, where most activities were coordinated by oral communication or by scribbled notes, some things unavoidably were done too late or simply forgotten. The solution for Køge was comprehensive use of electronic white boards.

Generative mechanisms: The solution was developed in a cooperative project by the hospital staff, the vendor of the white boards and the University of Roskilde (Hertzum and Simonsen, 2010). It was typically lightweight, it took the work processes as the starting point, and it was mainly developed by using standard technology (white boards), which was configured in an iterative and explorative way.

Adoption was voluntary, but the solution was very visible: the boards are large flat-screens that are hung in most rooms of the emergency department. The content is primarily *lists of patients*, which are continually updated. For example, in the reception area there is a *waiting list*, where names are listed, with expected waiting time. In the initial screening room there is a *prioritisation list*, which is used to allocate doctors. The screens are touch-based; the nurses re-allocate a patient by simply moving her/him in the list. A doctor may order a lab test by touching a field to the right of the patient’s name, and choose the test. Later, when the test is ready from the lab, a symbol will appear on the same field.

Outcome: The solution was perceived as quite successful. It did include some integration with heavy-weight IT, such as the lab connection, but it would have worked without it. Scaling to other departments may be more challenging, since its design was closely tailored to the work processes.

Case 4: Kasama City

Context: Kasama is a city north of Tokyo, with 79.000 inhabitants, 19.000 of these are over 65 years. In this city Hitachi, a giant corporation, has implemented a large IT solution with cloud technology, “Care and Medical Check-up Cloud”, being operative from March 2014.

Generative mechanisms: The solution was basically a patient journal system, adapted and extended to also include welfare services. It integrates the information from all services to the elderly: authorities, health personnel, patients, relatives, ambulance services, hospitals, pharmacies, care managers, service operators, and social workers (based on role based access). Patients and residents have access, and can register data. The users are reported to be satisfied, with the exception of pharmacists. The senior engineer at Hitachi explained:

“We want to improve the service level for residents, with seamless medical, care and prevention services based on shared information. The vision for further extension of our system includes the introduction of apps and connected sensors, and to offer statistics and BI solutions to the municipality and residents.”

According to Hitachi, the system will offer APIs for third party developers, and the plan is to implement the system in 40% of the market in greater Tokyo within 2 years.

Outcome: This may be seen as a typical example of heavyweight generativity; the solution was built on an existing system, and offered a broad range of new features to new user groups. It was developed

by IT specialists, who used the established infrastructure to integrate new components. It remains to be seen whether the solution will succeed as a platform for 3rd party innovators, but it will certainly be technically feasible.

5 Case Analysis

The four cases were all successful and forward-looking examples of e-health innovation. Considering the causes of success, a key finding is that generativity works quite differently in the lightweight versus the heavyweight cases. The main points are illustrated in table 3.

Generative mechanism	Heavyweight	Lightweight
Innovation	IT professionals recombine systems and middleware	Medical professionals and vendors recombine lightweight components with work tasks
Adoption	Mostly mandatory use, with organized implementation	Mostly voluntary use, where increased adoption generates more resources for the solutions
Scaling	Middleware software enables the linking of partner solutions	Limited, but successful scaling by replication

Table 3. Generative mechanisms of heavyweight and lightweight IT

Innovation

The innovation mechanism is based on the process of recombination (Yoo et al., 2012, Henfridsson and Bygstad, 2013), i.e. the space of possibility to recombine business and technical elements. Assessing the affordances offered by heavyweight and lightweight technology, we observe that they are rather different.

The affordances of heavyweight IT emerge from the interactions between different IT specialists, such as IT architects, integration specialist and IT managers, in co-operation with business managers. There are certainly some users in the background of heavyweight IT, but often they are represented by written requirements, or, as in agile projects, as GUI co-designers. Affordances are typically actualized through *integration*, i.e. creating new services by combining information from different systems, for example with middleware, as illustrated by the Integration Factory case, or by extending functionality as in the Kasama case.

In contrast, the affordances of lightweight IT emerge from the interactions of powerful users (such as doctors and managers) with IT product specialists. Affordances are typically actualized through the *combination* of a specific work task with a piece of standardized, but appropriated technology. For example, in the K ge case the combination of patient needs, medical personnel availability and the white-boards offers both doctors and patient the information for rational logistics. Similarly with the St.Hanshaugen case, the specialists and users are in the foreground, while the IT expertise is in the background.

Adoption

The adoption process is also different. Heavyweight projects tend to follow a waterfall model with installation of the solution, the training of users and the on-going support from IT staff. Use is often mandatory, both because the large investments, and because users are often dependent on the appropriate use of others. For example, in the Kasama case the pharmacists are dependent on the correct registration of patient data and doctors' prescriptions for the solution to work.

The adoption of lightweight IT is, in contrast, often voluntary and takes place in a more improvised process. In the St.Hanshaugen case, the tablet solution was initiated by the manager trying to solve a

problem, and discussing it with a vendor that was accidentally present. The first version of the solution was extremely simple, but as it proved useful it was gradually taken into use by more personnel. As the solution proved very useful, more resources were made available to increase the number of both services and more users. The Køge solution in Denmark was not completely voluntary, but its gradual diffusion was the result of a strong engagement from one of the care managers, who continuously improved and expanded the services. With a smile he commented, “you know, when the large whiteboards are in everybody’s sight all the time, they are hard to overlook.”

Scaling

Scaling is about extending networks in both user numbers and services. Heavyweight IT may scale well or poorly, depending on its architecture and the skills of the IT staff: for example, silo systems tend to scale poorly, because they are difficult to integrate with other systems. On the other hand, as the Integration Factory case illustrates, a specialised IT staff and advanced middleware technology can be a powerful mechanism for the scaling of separate systems. Likewise, the Kasama case shows the effective scaling of an electronic patient systems solution into another domain.

Scaling is more difficult with lightweight IT. The Køge solution, though impressive in terms of innovation and adoption, does not scale easily into other sites or domains, partly because it is dependent on some highly dedicated persons who made it work through tailoring it to a particular environment. The St.Hanshaugen solution also scales poorly, for different reasons; the number of links to one tablet is restricted to the number of patients the nurse can deal with. Thus, the next nurse must configure a new tablet solution from scratch, and so on. In this sense, we might say that the solution scales by replication, but it is of course an expensive approach.

Integration of heavyweight and lightweight IT?

Although both ecologies are generative there are in our four cases few examples of integration between lightweight and heavyweight IT. One exception is the integration at Køge between white boards and clinical systems, such as lab and imaging. This solution is particularly interesting, because it shows a way forward. First, it was initiated from the lightweight side, in order to provide the clinicians with details from the heavyweight systems. The lightweight solution would have worked without it, but it enhanced the benefits for the doctors. Second, it was conducted in a lightweight fashion, i.e. as a simple data exchange that was not requiring any changes in the heavyweight solution. Third, it was conducted as an incremental improvement, not as a development project.

6 Discussion: Generativity as Loose Coupling between Heavyweight and Lightweight IT

As the four cases illustrate, both knowledge regimes of heavyweight and lightweight are generative: Heavyweight IT is generative in the sense that the advanced architecture and specialised personnel constitutes a powerful resource for developing new services. Lightweight IT is generative in the sense that it allows the non-IT specialist to deploy, use and benefit from IT to support their work processes.

What is the generative relationship between heavyweight and lightweight IT?

To deal with this question we draw on the growing literature on platforms and the work on knowledge regimes of Howard-Grenville and Carlile (2006). We also build on the classical paper of Parnas (1972) on loose coupling. While heavyweight and lightweight IT has their internal generative capacities, it is their *interaction* that represents the real generative potential. While lightweight IT may appear more attractive in an innovation context, we should not romanticize lightweight IT. As DeLanda (1998) noted:

“After all, meshworks grow by drift and they may drift to places where we do not want to go. The goal-directedness of hierarchies is the kind of property that we may desire to keep at least for certain

institutions. Hence, demonizing centralisation and glorifying decentralization as the solution to all our problems would be wrong. An open and experimental attitude towards the question of different hybrids and mixtures is what the complexity of reality itself seems to call for”.

It is well documented that some of the most spectacular Internet success companies has built on platform architectures: Google was “built to build” (Iyer and Davenport, 2008) and Apple and Amazon have, in spite of their closed inner architecture, opened up a whole new market for apps and partner solutions. Platforms are mediating the activities of disaggregated clusters or ecosystems (Baldwin and Woodard, 2008). However, while platform architectures represent one excellent solution for the interaction between heavyweight and lightweight IT, they also have limitations; they tend to be built around one central actor, who controls the ecology, such as Apple and Google (Eaton et al., 2011). This may be beneficial in some settings, but in principle generativity should also work in more balanced relationship between heavyweight and lightweight IT.

Interaction, not integration

Since heavyweight and lightweight IT represent different knowledge regimes, the relationship is obviously non-trivial. Knowledge regimes tend to be more incompatible than often assumed, due to their nested structures of technology and work practices (Howard-Grenville and Carlile, 2006). For example, the nested structure of heavyweight IT is well documented in the software engineering literature (such as Sommerville et al., 2012), which describes in detail the interactions of software engineers and the various technology artefacts. The nested structure of lightweight IT, on the other hand, is most visible in the commercial and non-commercial descriptions of the affordances offered by the interaction of lightweight vendors and professionals.

It is therefore important here to differentiate between integration and interaction. Integration is a process where separate element become one whole, while interaction denotes the communication between separate entities. Integration increases technical (Parnas, 1972) and social complexity (Lawrence and Lorsch, 1967), and should be avoided when it is not really necessary. Since heavyweight and lightweight IT not only are different technologies, but different knowledge regimes, it is essential that we understand that generativity does not mean tight integration.

Design principles: Loose coupling

The potential lies in connecting the heavyweight and lightweight segment, in a way that is generative, i.e. without hindering the generative attributes of each of them. A well-known tactics to reduce complexity in software design is the principle of *loose coupling* between modules (Parnas, 1972). Loose coupling enables less interdependency, reduces information flow and requires less coordination.

The principle of loose coupling can be extended as more general guideline. In our context it means that the two knowledge regimes should be loosely coupled, not tightly integrated. In order to maintain their capacities to interact (DeLanda, 2009), we propose three design principles.

1. They should be loosely coupled technically.

Integration is a double-edged sword; it allows for tight co-operation, but it also increases complexity by establishing many dependencies. Integration between heavyweight and lightweight IT is complex and expensive. Using Howard-Grenville and Carlile’s terms, we (i) need to integrate very different technologies, in (ii) incompatible work processes and (iii) in incompatible discourses. Therefore, lightweight IT should support work processes successfully before it is integrated.

An illustrative example is the way the company Imatis worked in the Danish case; they started with implementing work process support in tight co-operation with the clinicians. Only after the white boards were supporting the processes in daily practice, they started to integrate the services with heavyweight technology. If they had started with integration, it is quite likely that the innovation process would have been slowed down and may even have halted. The project would also have faced a much larger initial cost, greatly increasing the risk of the initiative.

2. *They should be loosely coupled in terms of standards.*

It is tempting to assume that standardisation will ease the interaction between heavyweight and lightweight IT, for instance between a sensor and a patient journal systems. However, at this early stage of evolution standards are not necessarily the right solution (Hanseth and Bygstad, 2012). Standards are means, not aims. Recalling the St.Hanshaugen case, we noticed that the lack of standards were not a hindrance for their solutions. If companies are asked to “wait” until the standards are ready, it will effectively stifle innovation.

3. *They should be loosely coupled in terms of organisation.*

Heavyweight vendors will no doubt try to expand into the lightweight market, by extending their heavyweight solutions with lightweight services. Some of them will probably succeed, but the heavyweight culture is not well suited to this. Researchers such as Henderson and Clark (1990) and more recently Hylving et al. (2012) has shown that large product-oriented organisations tend to concentrate on a *dominant design* of their technology, which is also reflected in their internal organisation. This enhances product improvement, but not architectural innovation.

This implies that innovation is best served by different organisations developing heavyweight and lightweight IT. One way to do this would be for heavyweight vendors to offer their solutions as platforms for 3rd party innovators, following the Apple trail. Another model could be the more advanced solution demonstrated by the Integration Factory, where *thin calls* from lightweight are routed into the heavyweight domain. As these examples show, the mechanisms for interaction are in itself an innovation area.

Looking forward

This research is conceptual and exploratory, and the intention is to open up a new perspective for investigating a new trend in IT-based service innovation. At this point we suggest two avenues of future research; first we need more empirical studies of innovation processes encompassing both heavyweight and lightweight IT. Second, for the IT departments around the world, the governance of lightweight IT is basically unsolved, and we need to develop new, practical models that combine the generative potential of heavyweight and lightweight IT.

7 Conclusion

This paper proposes a simple terminology for understanding and dealing with two current, and extremely complex, phenomena; the evolution of heavyweight IT and the emergence of lightweight IT. We regard them not as merely different technologies, but as different knowledge regimes. They are, however, mutually dependent of each other, in the sense that heavyweight IT can work as a platform for innovative lightweight IT, and that lightweight IT offers an arena for innovation that is outside the scope of heavyweight IT.

Building on the theoretical lens of generativity, we offer two contributions. First, we show that generativity in digital infrastructures is supported by the interaction of loosely coupled heavyweight and lightweight IT. Second, we offer a set of design principles that respects the particular generativity of each paradigm, but offers a way forward to exploit the generative potential of interaction.

Acknowledgements

We thank the many informants for their time and engagement, and ECIS reviewers. This research was part of the FIGI project, supported financially by the Regionale Forskningsfond, Hovedstaden.

References

- Alemdar, H. and Ersoy, C. (2010). "Wireless sensor networks for healthcare: A survey." *Computer Networks*, 54(15): 2688–2710.
- Avital, M. and Te'eni, D. (2009). "From generative fit to generative capacity: Exploring an emerging dimension of information systems design and task performance". *Information Systems Journal*, 19, 345-367.
- Baldwin, C. and Woodard, J. (2008). "The Architecture of Platforms: A Unified View". Working Paper, Harvard Business School. http://www.hbs.edu/faculty/Publication%20Files/09-034_149607b7-2b95-4316-b4b6-1df66dd34e83.pdf
- Bannister, F. (2001). "Dismantling the silos: extracting new value from IT investments in public administration". *Information Systems Journal*, 11(1):65-84.
- Bergvall-Kåreborn, Birgitta and Howcroft, Debra (2011) "Mobile Applications Development on Apple and Google Platforms," *Communications of the Association for Information Systems*: Vol. 29, Article 30.
- Bouwman, H.; van Houtum, H.; Janssen, M; and Versteeg, G. (2011) "Business Architectures in the Public Sector: Experiences from Practice," *Communications of the Association for Information Systems*: Vol. 29, Article 23.
- Campbell, J. and Pedersen, O.K. (2006). "Knowledge Regimes and Comparative Political Economy". In Daniel Béland and Robert Henry Cox (eds.), *Ideas and Politics in Social Science Research*. New York: Oxford University Press.
- Christensen, C.M. (2009). *The Innovator's Prescription. A Disruptive Solution for Health Care*. New York, McGrawHill.
- DeLanda, M. 2006. *A New Philosophy of Society: Assemblage Theory and Social Complexity*. New York, NJ: Continuum.
- DeLanda, Manuel (1998) "Meshworks, hierarchies and interfaces". In: Beckman, John (ed.) *The virtual dimension. Architecture, representation and crash culture*. Princeton, NJ: Princeton Architectural Press, 274–285.
- Eaton, B., Elaluf-Calderwood, S., Sørensen, C, and Yoo, Y. (2011). "Dynamic structures of control and generativity in digital ecosystem service innovation: the cases of the Apple and Google mobile app stores". Working paper of the London School of Economics. <http://is2.lse.ac.uk/wp/pdf/wp183.pdf>
- Elder-Vass, D. (2007). "A Method for Social Ontology: Iterating Ontology and Social Research". *Journal of Critical Realism*, 6(2).
- Etzori, L., Iera, A. and Morabito, G. (2010). "The Internet of Things: A survey". *Computer Networks* 54:2787–2805.
- European Commission (2011). "European countries on their journey towards national eHealth infrastructures". http://www.ehealth-strategies.eu/report/eHealth_Strategies_Final_Report_Web.pdf.
- Gartner (2014) "Bimodal IT: How to Be Digitally Agile Without Making a Mess". Gartner.com.
- Greenhalgh T, Stramer K, Bratan T et al. (2010). "Adoption and non-adoption of a shared electronic summary record in England: a mixed-method case study". *British Medical Journal*, 340.
- Harris, J, Ives, B. and Junglas, I. (2012). "IT Consumerization: When Gadgets Turn Into Enterprise IT Tools". *MIS Quarterly Executive*, 11(3): 99-112.

- Hanseth, O., and Lyytinen, K. (2010). "Design theory for dynamic complexity in information infrastructures: the case of building internet." *Journal of Information Technology*, 25(1), 1-19.
- 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.
- Henfridsson, O. and Bygstad, B. (2013). "The Generative Mechanisms of Digital Infrastructure Evolution". *MIS Quarterly*, 37(3):907-931.
- Iyer, B. and T. H. Davenport (2008). Reverse Engineering. Googles Innovation Machine. *Harvard Business Review* April: 59-68.
- Hertzum, M. and Simonsen, J. (2010). Effects-driven IT development: an instrument for supporting sustained participatory design. Proceedings of the ACM 11th Biennial Participatory Design Conference, 61-70.
- Hylving, L, Henfridsson, O, Selander, L. (2012). "The Role of Dominant Design in a Product Developing Firm's Digital Innovation". *Journal of Information Technology Theory and Application*, 13(2): 5-21.
- Lane, D. A. (2011). "Complexity and Innovation Dynamics", in C. Antonelli, (ed.), *Handbook on the Economic Complexity of Technological Change*. Edward Elgar Publishing Ltd.
- Lawrence, P.R. and Lorsch J.W. (1967). "Differentiation and integration in complex organizations". *Administrative Science Quarterly*, 12(1):1–47.
- Leonardi, P. M. (2011) "When Flexible Routines Meet Flexible Technologies: Affordance, Constraint, and the Imbrication of Human and Material Agencies" *MIS Quarterly*, 35(1): 147-167.
- Majchrzak, A. and Markus, M. L. (2012). Technology Affordances and Constraints in Management Information Systems (MIS). *Encyclopedia of Management Theory*, (Ed: E. Kessler), Sage Publications.
- Mccarthy, D., Mueller, K. and Wrenn, J. (2009). "Kaiser Permanente: Bridging the Quality Divide with Integrated Practice, Group Accountability, and Health Information Technology". <http://www.commonwealthfund.org>
- Miorandi, D. et al.(2012). "Internet of things: Vision, applications and research challenges". *Ad Hoc Networks* 10(7):1497–1516
- Niehaves, B., Sebastian Köffer, and Kevin Ortbach, (2012)"IT Consumerization – A Theory and Practice Review. *AMCIS Proceedings*. Paper 18.
- Parnas, D.L. (1972). "On the Criteria To Be Used in Decomposing Systems into Modules". *Communications of the ACM* 15 (12): 1053–58.
- Phelan, S. E. (2003). "What is complexity science really?" *Emergence*, 3(120-136).
- Rai, A., Venkatesh, V., Bala, H. and Lewis, M. (2010). "Transitioning to a Modular Enterprise Architecture: Drivers, Constraints, and Actions". *MIS Quarterly Executive* 9(2):83-94.
- Rosen, M., Lublinsky, B., Smith, K. T. and Balcer, M. J. (2008). *Applied SOA: Service-Oriented Architecture and Design Strategies*. Indianapolis, Wiley Publishing.
- Ross, J.W., Weill, P. and Robertson, D. (2006). *Enterprise Architecture as Strategy: Creating a Foundation for Business Execution*, Harvard Business School Press.
- Sauer, C. and Willcocks, L.P., "Unreasonable expectations – NHS IT, Greek choruses and the games institutions play around mega-programmes". *Journal of Information Technology*, 2007. 22: p. 195-201.

Sommerville, I., Cliff, D., Calinescu, R., Keen, J., Kelly, T., Kwiatkowska, M.Z., McDermid, J.A., Richard F. and R.F. (2012). "Large-scale complex IT systems". *Communications of the ACM* 55(7): 71-77.

Sherer, Susan A. (2014) "Patients Are Not Simply Health IT Users or Consumers: The Case for "e Healthicant" Applications," *Communications of the Association for Information Systems*: Vol. 34, Article 17.

Smith, J.E.H. (2011). *Divine Machines: Leibniz and the Sciences of Life*. Princeton University Press.

Volkoff, O. and Strong, D. (2013). "Critical Realism and Affordances: Theorizing IT-Associated Organizational Change Processes". *MIS Quarterly* 37(3):819-834.

Yoo, Y., Boland, R.J., Lyytinen, K., & Majchrzak, A. (2012). "Organizing for innovation in the digitized world". *Organization Science*, 23(5), 1398–1408.

Yoo, Y. (2013). "The Tables Have Turned: How Can the Information Systems Field Contribute to Technology and Innovation Management Research?" *Journal of the Association for Information Systems*, 14(5): 227-236.

Zittrain, J. L. (2006). "The Generative Internet." *Harvard Law Review*, 119, 1974-240.