Journal of Information Technology (2017) 32, 180–193

www.palgrave.com/journals

Research Article

Generative innovation: a comparison of lightweight and heavyweight IT

Bendik Bygstad[1,2]

[1]Department of Informatics, University of Oslo, Oslo, Norway;
[2]Norwegian School of Economics, Bergen, Norway

Correspondence:
B Bygstad, Department of Informatics, University of Oslo, Ole Johan Dahls Hus, Gaustadalléen 23, Oslo 0737, Norway.
E-mail: bendikby@ifi.uio.no

Abstract
This paper proposes a simple terminology for understanding and dealing with two current phenomena; we suggest calling them heavyweight and lightweight IT. Heavyweight IT denotes the well-established knowledge regime of large systems, developing ever more sophisticated solutions through advanced integration. Lightweight IT is suggested as a term for the new knowledge regime of mobile apps, sensors and bring-your-own-device, also called consumerisation and Internet-of-Things. The key aspect of lightweight IT is not only the cheaper and more available technology compared with heavyweight IT, but the fact that its deployment is frequently done by users or vendors, bypassing the IT departments. Our theoretical lens is generativity, the idea that complex phenomena arise from interactions among basic elements. In the context of IT, generativity helps to explain the creative potential of flexible digital technology for knowledgeable professionals and users. The 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 enfolds 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.
Journal of Information Technology (2017) **32**, 180–193.
doi:10.1057/jit.2016.15; advance online publication, 24 May 2016
Keywords: Lightweight IT; heavyweight IT; generativity; digital infrastructures; e-health

Introduction

In June 2014 Apple announced a partnership with EPIC, a vendor of Electronic Patient Journal systems, who covers around 50% of the American population's medical records. 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 third-party vendors called HealthKit. The partnership created a lot of interest, and commentators (Munro, 2014) pointed at the disruptive potential of the alliance; while EPIC's attention and competence are focused on clinicians' needs, Apple's products are the preferred devices for millions of patients and their families. Moreover, the iTunes platform has triggered an astonishingly innovative arena of mobile apps (Eaton *et al.*, 2011). Together, the alliance might be an e-health game changer, but from a research perspective it also presents some salient issues that are relevant beyond the health sector.

The partnership vividly illustrates that two current trends are changing the IT industry and the way we use IT. First, the IT systems within many sectors – banking, travel, public administration, retail and health – are growing in size and interconnectivity. In particular, 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. The solutions are quite advanced, but also more complex (Sommerville, 2010; Sommerville *et al.*, 2012).

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 IT solutions, at different 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 (Atzori *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, 2006). However, some researchers in the IS field, such as Howard-Grenville and Carlile (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). In our context we take this to include the network of key actor groups (such as IT professionals, users, vendors), work practices, certain technologies and the shared knowledge on the appropriate development and use.

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 IT*,[1] 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 regards the two trends as a paradigmatic shift to two different knowledge regimes: *heavyweight and lightweight IT*. Heavyweight IT is here defined as a knowledge regime, driven by IT professionals, enabled by systematic specification and proven digital technology, and realized through software engineering. 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 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 solutions, enabled by the consumerisation of digital technology, and realized through innovation processes.*

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, 2009). At the same time it is also characterised 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 healthcare 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 healthcare area. These solutions are quite diverse; some of them support work processes for healthcare 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 heavyweight 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?

Our contribution is to develop a simple yet powerful set of concepts to shed light on a current phenomenon. As explained by Glaser and Strauss (1999) categories should be sensitising – yield a meaningful picture that lay people can grasp and relate to their experiences. We proceed by defining heavyweight and lightweight IT. We frame our analysis within the *digital infrastructure*[2] research (also called information infrastructures or cyber-infrastructures), which is primarily interested in the dynamics of large networks rather than stand-alone applications (Hanseth and Lyytinen, 2010). To develop our argument we draw on the literature on generativity, that is, the ability of technical and social elements to interact and recombine to produce or expand 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.

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 enterprise resource planning (ERP) and other transaction systems, based on databases servers and integration software, such as enterprise bus frameworks (Rosen *et al.*, 2008) and service-oriented architecture (Erl *et al.*, 2015). 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

Table 1 Heavyweight and lightweight IT

	Heavyweight IT	*Lightweight IT*
	A knowledge regime, driven by IT professionals, enabled by systematic specification and proven digital technology and realized through software engineering	A knowledge regime, driven by competent users' need for solutions, enabled by the consumerisation of digital technology and realized through innovation processes
Profile	Back-end: Supporting documentation of work	Front-end: Supporting work processes
Owner	IT department	Users and vendors
Systems	Transaction systems	Process support, apps, BI
Technology	PCs, servers, databases, integration technology	Tablets, electronic whiteboards, mobile phones
IT architecture	Fully integrated solutions, centralised or distributed	Non-invasive solutions, frequently meshworks (heterogeneous networks)
Development culture	Systematics, quality, security	Innovation, experimentation
Problems	Increasing complexity, rising costs	Isolated gadgets, security
Discourse	Software engineering	Business and practice innovation

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*, that is, 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, that is, 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 user interfaces, 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 emergence of Robotic Process Automation (RPA) (Lacity and Willcocks, 2015), which allows non-IT specialist to implement service automation tools in white-collar work processes. Table 1 summarises the attributes of heavyweight *vs* lightweight IT.

The distinguishing feature is not the technology in itself, but the *digital infrastructure*; the network of technology, designers and users, kept together by a knowledge regime. The profile of heavyweight IT is typically back-office systems, such as enterprise systems in production and retail, accounting systems in finance and and Electronic Patient Record (EPR) systems in healthcare. Lightweight IT typically supports the immediate needs of the user, providing process support (for instance when aircraft pilots use tablets for checklists) or simply provides a single piece of information, such as the bodily pulse on your Apple Watch.

Heavyweight IT uses proven technology, such as servers and databases, integrated by advanced solutions, in well-designed architectures, which may be centralised, distributed or delivered by cloud services. It is owned by a specialised IT unit; often the IT department of a large organisation. Developing heavyweight technology requires specialised IT competence, focusing on requirements, reliability and security. In contrast lightweight IT is usually developed as *non-invasive* solutions, acting on the presentation layer, while heavyweight solutions interact with the business logic and data access layers (Lacity and Willcocks, 2015). Lightweight IT uses consumer technology such as smart phones, tablets, apps and whiteboards, and operates largely outside heavyweight IT resources. Lightweight IT is usually initiated by users in cooperation with a specialised vendor, for example, when a doctor have an idea of a new information service for a patient group, and finds a vendor who can deliver this. Because of the bottom-up approach lightweight architectures are usually meshworks; they grow organically as user needs change. The development culture is innovation and experimentation; development cycles are short, and product life cycles may also be short.

Heavyweight IT has a thorny history of failures, but is today a mature field with a strong foundation in software engineering (Sommerville, 2010). However, with increasing scale and integration, costs are escalating and technical stability is challenged by increasing complexity (Sommerville *et al.*, 2012). Lightweight IT has another set of problems; it easily becomes isolated gadgets with poor integration, and particularly in healthcare, privacy and security issues arise quickly.[3]

It should be underscored that heavyweight and lightweight IT are *ideal types* (used for analytical purposes, to help us sort out things in a complex world), and in practice there are examples that might be in between, or not fit in the simple dichotomy. Can lightweight IT become heavyweight? Yes, this happens frequently, for example, when a lightweight application is fully integrated with heavyweight solutions, and subjected to heavyweight governance. We will discuss this issue in more depth in the section 'Discussion: Generativity arise from interactions between heavyweight and lightweight IT'.

The IT health sector is currently in the middle of an IT revolution (Christensen, 2009). 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 use advanced solutions with specialized IT staff. They are also very costly, but promise to solve key parts of the silo problem of the past.

In healthcare the lightweight segment is a thriving arena of intense innovation, ranging from bodily sensors to Internet tablets with Facebook for the elderly (Sherer, 2014). It still makes sense to characterise these solutions as lightweight, because they are usually initiated by medical practitioners or patients, 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 that the development cultures differ substantially; heavyweight IT is increasingly being professionalized while lightweight IT is becoming a large innovation arena. The explosive growth of third-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 and integrated solutions designed by enterprise architects, while lightweight IT often has emerged as what DeLanda (1998) has called *meshworks*, that is, 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.

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. There is a parallel to earlier paradigm shifts in IT; in the 1980s the PCs fundamentally changed the user experience, but still relied heavily on central computers; in the 1990s the World Wide Web revolutionised the linking of information, but still relied heavily on the installed base of networked PCs.

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 Woodard, 2008). Current software ecosystems consist of platforms, their interfaces and apps that connects to the platforms (Tiwana, 2014).

Platforms can support third-party innovation, and some platform owners realise 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 should maintain a delicate balance of control; if the control of third-party contributors becomes too tight then the ecosystem loses its ability to generate innovation. Reversely, if the platform is too malleable, it easily becomes too fragmented to serve as a platform. Ghazawneh and Henfridsson (2015) proposed a typology of platform-based marketplaces, in terms of openness: closed, censored, focused and open.

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.

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).

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, which includes the user and developer communities involved. Lane (2011) argued that a community's capacity to innovate is determined by its *generative relationships*, that is, the ongoing discourse between the actors on their interpretation and use of artifacts. Likewise, Avital and Te'eni (2009) 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.

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. 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 (2013), Leonardi (2011) and Sørensen *et al.* (2015), 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[4] (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

A successful digital infrastructure is self-reinforcing, that is, it recursively feeds on itself. The innovation mechanism in

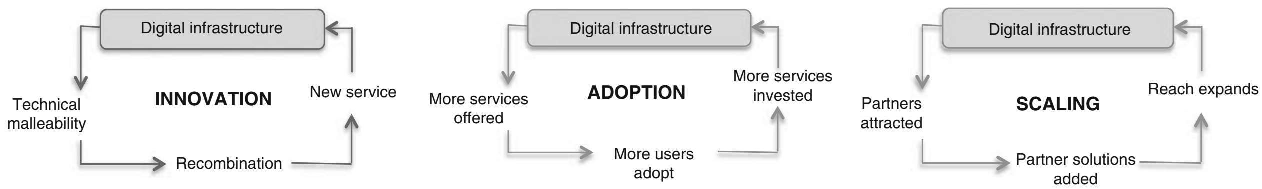

Figure 1 Generative mechanisms.

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. Adoption is self-reinforcing in that increasing use triggers investment into new services, which leads to more use. And scaling is self-reinforcing through the recruitment of partners, which increases the attractiveness of the infrastructure, which attracts more partners.

The three mechanisms may also reinforce each other: Adoption leads 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.

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 networks 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.

Research approach

The study was part of a larger research programme within e-health. Two longitudinal studies were conducted; an e-health megaproject in Norway (Bygstad and Hanseth, 2015) and a study of welfare technologies in Norway and Japan (Bygstad and Lanestedt, 2014). The research design of the two programmes did not include the concepts of heavyweight and lightweight IT, which emerged as part of the data analysis.

The conceptualization of heavyweight and lightweight IT developed over a period of 2 years, through an iterative process of fieldwork, literature study and theory development. It started with a longitudinal study of an e-health megaproject, which was characterised by an almost overwhelming organisational and technical complexity, standardising processes and technology in 13 hospitals (each with several geographically dispersed units), enabled by advanced integration, to a total cost of around a billion Euros. Then, in 2013–2015 a comparative study of welfare technologies in Japan and Norway was conducted, which included large and complex systems, but also some more improvised, but well-working solutions with simple and cheap technology such as tablets. See Table 2 for an overview.

Table 2 Steps in the research process

step	*Activity*	*Insights*
1	Longitudinal study of e-health mega-project (2009–2015)	Large-scale systems integration is feasible with new technology, but also demanding and costly
2.	Comparative study of welfare technologies in Norway and Japan (2013–2015)	Welfare IT solutions are realised both through advanced, integrated solutions, but also with simple solutions based on commercially available technology
3	Concept development	Heavyweight and lightweight IT are different knowledge regimes
4	Concept validation	Generativity works differently in heavyweight and lightweight IT
5	Theory development	Heavyweight and lightweight IT should be loosely coupled, technically and organisationally

The concepts of heavyweight and lightweight IT emerged from a meeting with a successful entrepreneur in e-health technology, Morten Andresen of Imatis, as we discussed hospital and welfare IT solutions. He commented that coming from the lighter side of technology he (and other start-up firms) had great difficulties in cooperating with the large systems vendors, because of different perspectives on IT:

> We work pretty differently. We start with the clinical work processes, and try to understand how to support them with IT, while the large vendors start with their products and their uses for clinicians. We prioritize short-term usefulness, and think about integration and security later, while the large vendors do the reverse. We feel that they are a hindrance for innovation, while they probably think that we are a bunch of irresponsible hackers.

From there we started to group our cases into heavyweight and lightweight, and realised that the differentiating criteria were not technology, but the knowledge regime, that is, the overall approach to how IT can be used in work practices, and the collective conventions on the appropriate use (Howard-Grenville and Carlile, 2006). We found reasonable support in IS research for this categorisation, in particular from studies on consumerisation (Niehaves *et al.*, 2012), platforms (Baldwin and Woodard, 2008) and Internet-of-Things (Atzori *et al.*, 2010). From the portfolio of many sub-projects in the two research programmes, four cases were chosen to validate the concepts, and to serve as a basis for theory development. We also identified generativity as a fruitful theoretical lens (Henfridsson and Bygstad, 2013).

Data collection

The cases (Table 3) were selected from a larger portfolio of cases in the two research programs, 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 reasonably successful healthcare projects; they leveraged available technology to a specific purpose, were competently managed and the solutions were taken into use. They were also small enough to warrant a relatively detailed analysis of generativity, but they were also parts of larger initiatives and infrastructures.

Interviewee details are documented in the Appendix. From our data bank of the two research programs we drew on data 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 selected information on the actors and events, and the recombination of technical and social elements. For instance, we investigated the interplay of managers, developers, users and vendors, and documented in detail how a new service was conceptualised, who was involved and how the development enfolded
- 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 selected 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

Data analysis

The two last steps in the study (Table 2, Steps 4 and 5) were conducted as follows. 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.

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.

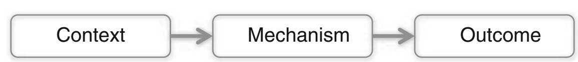

Figure 2 The CMO scheme.

Table 3 Cases

Case period	*Organisation*	*Type*	*Data collection*
1. The Integration Factory (2012–2016)	Regional Authority, Oslo, 80,000 employees	Heavy-weight	*Interviews:* IT managers, developers, architects, doctors, nurses (18) *Observations*: Demo of integrated solution *Documents:* Strategy, IT architecture document, process description
2. St.Hanshaugen Elderly Care (2014–2015)	Oslo Municipality, Norway, 100,000 employees	Light-weight	*Interviews:* Care manager, nurses, vendor representatives, vendor integrator (7) *Observations*: Demo of tablet solution *Documents:* Product description, vendor documentation
3.Køge Hospital Køge (2014–2015)	Region Sealand, Denmark, 14,000 employees	Light-weight	*Interviews:* Intensive care manager, nurses, developers, vendor rep (10) *Observations*: Observing use of whiteboard solutions in several units *Documents:* IT architecture, product descriptions, process description. The case is also documented in Hertzum and Simonsen (2010)
4. Kasama City Welfare services (2014)	*Kasama City*, Japan, 76,000 inhabitants	Heavy-weight	*Interviews*: Ministry of Health officials, vendor managers, architect, developers, nurses and elderly care personnel (7) *Observations:* Demo of solution, observation of solution in use *Documents:* IT architecture, product description

Table 4 Case outcomes

Case	*System quality*	*Information quality*	*Use*	*User satisfaction*
1. Integration Factory	High	High	High	(Not end-user solution)
2. St. Hanshaugen Elderly Care	Medium	Medium	High	High
3. Køge Hospital	High	High	High, but variable	High
4. Kasama City Welfare services	High	High to medium	In pilot phase	High in pilot phase

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 4.

Cases

The four cases were all public sector projects in healthcare, with private vendor technologies. They were all successful, but with some variations on the DeLone and McLean (2003) criteria, as shown in Table 4.

As illustrated in Table 4 the outcomes varied. The first case was a large technical integration solution with no end-users, while the second case presented a simple lightweight solution with satisfied users. The third case included an advanced lightweight solution for a hospital, while the fourth presented a large integrated solution for a whole city, in a pilot phase.

Case 1: the integration factory

Context

In 2012, the South-East 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, that is, 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 BizTalk solution enables routing and transformation of massages between the EPR system (DIPS) and the other systems (Fagsystem) within the Oslo University Hospital, and with systems in other hospitals (via the Gateway) and other registers.

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 ongoing 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 became the lynchpin of the large e-health programme, and was a key factor in the successful implementation. The integration solution, enabled by the Integration Factory, worked 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 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 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

The national strategy for elderly care is based on the premise that the elderly prefer to live in their homes as long as possible. St. Hanshaugen 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 (for example) 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

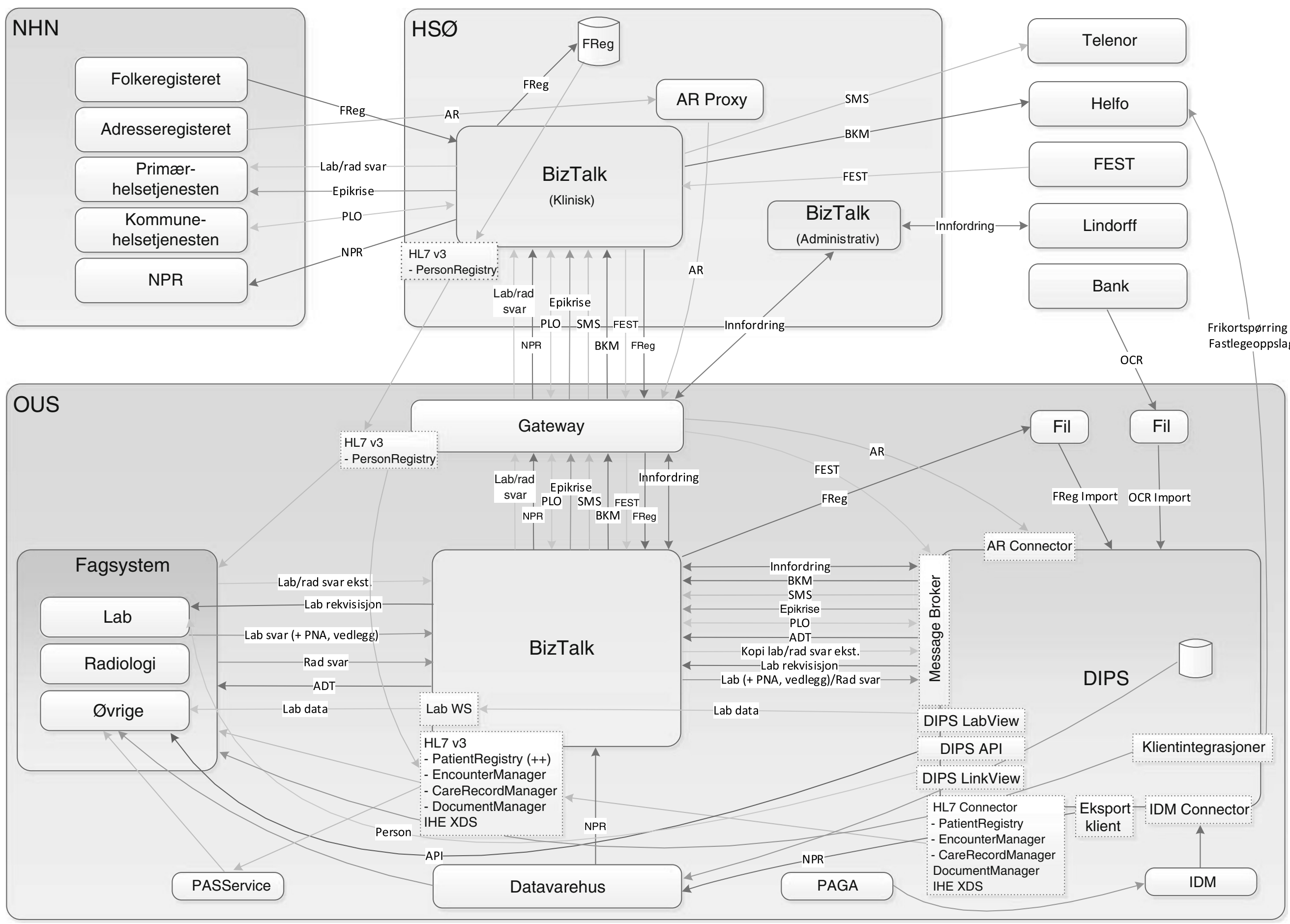

Figure 3 The integration platform at Oslo university hospital
Source: Digital renewal programme.

solutions and standards) to the nurse's tablet. It is possible to relay messages to family members.

Outcome

The solution solved a key problem for the communication between elderly living at home and the municipal care unit, at a low cost. 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 with 36,000 inhabitants, lying south of Copenhagen. Our case was the emergency department of the hospital, 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 whiteboards.

Generative mechanisms

The solution was developed in a cooperative project by the hospital staff, the vendor of the whiteboards 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 (whiteboards with software from Imatis), 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

Table 5 Generative mechanisms of heavyweight and lightweight IT

Generative mechanism	*Heavyweight IT*	*Lightweight IT*
	Cases: Integration Factory and Kasama City	Cases: St.Hanshaugen and Køge Hospital
Innovation	IT professionals recombine systems and middleware	Medical professionals and vendors recombine lightweight components with work tasks
Adoption	Mostly mandatory use, with organised 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

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, and was used extensively by all employees. It did include some integration with heavyweight 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 local work processes.

Case 4: welfare services in Kasama city

Context

Japan's approach to infrastructures has been to let corporations build, own and run them commercially. 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 Japanese corporation, has implemented a large IT solution for welfare services with cloud technology, 'Care and Medical Check-up Cloud', being operative from March 2014. The solution covers 5 hospitals, 40 clinics, 32 dentists and 50 care stations for the elderly.

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 through web interfaces, and can register their own data.

This may be seen as a typical example of heavyweight generativity; the solution was built on an existing EPR system, and offered a broad range of new features to new user groups. It was developed by IT specialists, who used the established (and proprietary) infrastructure to integrate new components.

Outcome

The roll-out of the solution was successfully done in 2014, and services gradually taken into use. The users are reported to be satisfied, with the exception of pharmacists, who previously had a more specialised system. The senior engineer at Hitachi explained:

> We wanted 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.

The system will offer APIs (application program interfaces) for third-party developers, and the plan is to implement the system in 40% of the market in greater Tokyo within 2 years. It remains to be seen whether the solution will succeed as a platform for third-party innovators, but it will certainly be technically feasible.

Case analysis

The four cases were all reasonably 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 *vs* the heavyweight cases. The main points are illustrated in Table 5.

Innovation

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

Innovations 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 user interface co-designers. New services are typically actualised though *integration*, that is, 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 opportunities for innovations of lightweight IT emerge from the interactions of powerful users (such as doctors and managers) with IT product specialists. New services are typically actualised through the *combination* of a specific work task with a piece of standardised, but appropriated technology. For example, in the Køge case the combination of patient needs, medical personnel availability and the whiteboards 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 ongoing 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 representative who 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 whiteboards 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, that is, 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.

Discussion: generativity arise from interactions 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 (Baldwin and Woodard, 2008) and the work on knowledge regimes of Howard-Grenville and Carlile (2006). We also build on the classical papers of Parnas (1972) and Weick (1976) 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 romanticise 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.
>
> (p. 285)

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). Platform architectures represent an excellent solution for the interaction between heavyweight and lightweight IT. For instance, in the case of the Kasama solution, the vendor Hitachi was planning to open their heavyweight solution to third-party app developers, who could offer end-user solutions for specific purposes.

However, platforms 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

As heavyweight and lightweight IT represent different knowledge regimes, the relationship is obviously non-trivial. Knowledge regimes tend to be more incompatible than often assumed, because of 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 advanced technical and organisational solution of the Integration factory illustrates this point. The nested structure of lightweight IT, on the other hand, is most visible in the interaction of lightweight vendors and professionals, focusing on work processes, as illustrated in the case of Køge Hospital in Denmark.

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 organisational complexity (Weick, 1976), 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, that is, 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. We propose three design principles.

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 need to integrate very different technologies, in (i) incompatible work processes and (ii) 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 whiteboards 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.

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, 2015). Standards are means, not aims. Recalling the St. Hanshaugen case, we noticed that the lack of standards were not a hindrance for their solutions. When asked about this issue, the CEO of the small vendor Dignio responded:

> I certainly know that the authorities are very concerned about standards and interoperability, assuming that this is a prerequisite for interacting solutions. Frankly, in our experience this is not a problem. When we are asked to connect a new device, such a sensor or a tablet, our experience is that the software of the device can deliver its output in many formats and protocols, and the receiving system can also handle a range of formats. For our engineers this is just plain configuration.

This is not to argue that standards are unnecessary, but it will probably take decades to standardise all elements of the e-health field because of its complexity. If companies are asked to 'wait' until the standards are ready, it will effectively stifle innovation.

They should be loosely coupled in terms of organisation.

Weick (1976) showed that loose coupling in organisations have several advantages, such as enabling units to be more robust to external changes and breakdowns, supporting local adaptation and also less expensive, since it has less need for coordination. All these aspects are relevant for our third design principle.

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 third-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.

The consequences of loose coupling

Willcocks *et al.* (2015) argue that lightweight IT should be subjected to the same governance principles as heavyweight, because of the risks of becoming 'shadow IT' with the associated problems of poor reliability, organisational alignment and security. While this position is well argued we tend to disagree, for two reasons.

First, lightweight IT is in an early phase, and users and vendors are experimenting with technology, solutions and organisational mechanisms to foster innovation. An important part of the scene is the Internet and mobile giants with their platforms, such as Google, Apple and Samsung, who are both enabling and monopolising emerging ecologies (Eaton *et al.*, 2011). By definition innovation processes are hard to plan; the solutions of the future are not only hard to plan but also hard to envision, and emerge through interactions of diverse actors. That is why a third-party ecology is more innovative than a heavyweight IT department. Therefore, it can be argued that a strict governance of lightweight IT runs a high risk of taking the wrong decisions. That is not to say that chaos should be allowed to develop. Rather, following Pollock *et al.* (2007) the argument is that of *generification*; as specific lightweight solutions gradually stabilise and become widely used they should (also gradually) be subject to standardisation and centralised governance, in order to scale technically and economically. In this process they might become heavyweight.

Second, heavyweight IT is currently overloaded in most organisations, with rising cost, long backlogs and increasing complexity (Sommerville *et al.*, 2012). Adding lightweight IT into this situation will hardly make things easier. Rather, there is a need for a sensible division of labour between heavyweight and lightweight IT; heavyweight IT should concentrate on the stable elements of digital infrastructures, such as basic registers and networks, security mechanisms and shared business components. This is the core of the heavyweight knowledge regime. Complementary, lightweight IT should provide the unstable and short-lived elements of the infrastructures: end-user interfaces, solutions for specific needs, Business Intelligence, RPA and similar solutions. For this division of labour to work, it is essential that the two knowledge regimes be loosely coupled.

Limitations

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. We acknowledge that there are many issues related to the interplay of heavyweight and lightweight IT that were not dealt with here. In particular, the issue of IT security and privacy is a real challenge for lightweight IT, and should of course not be taken lightly. The same applies to IT governance. These topics, however, were too extensive to be dealt properly with here.

Our analysis built on four cases from the healthcare sector. While we see no reason that our findings should not be valid and useful outside the healthcare sector, further research is needed to investigate this.

Conclusion

This paper proposes a simple terminology for understanding and dealing with two current and 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 more 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. The practical design implication is that heavyweight and lightweight IT should be only loosely integrated, both in terms of technology, standardisation and organisation.

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. Gartner's (2014) concept of *bimodal IT* is useful for organising the IT department, but it does not address the full challenge of lightweight IT.

Acknowledgements

The author would like to thank the *JIT* reviewers for useful feedback, and in particularly the author would also like to thank the Senior Editor for her engagement with the paper. An earlier version of this paper was published at ECIS 2015. Thanks are also due to the many informants for their time and engagement. This research was part of the FIGI project, supported financially by the Regionale Forskningsfond, Hovedstaden.

Notes

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.

2 Infrastructure is used as a term to conceptualise interconnected system collectives. The past 20 years have witnessed research on digital infrastructures covering different settings such as health, telecom, finance, government and manufacturing. Hanseth and Lyytinen (2010) defined an information infrastructure as 'a shared, open (and unbounded), heterogeneous and evolving socio-technical system (which we call installed base) consisting of a set of IT capabilities and their users, operations and design' (p. 4).

3 The security and privacy issues of lightweight IT are indeed problematic, and must be addressed, theoretically, technically and organisationally. They are however not dealt with in this paper, which focuses on the generativity aspects.

4 Following Bhaskar ([1975] 1997), we define generative mechanisms as causal structures that generate observable events. The outcome of a mechanism is contingent, that is, it may vary depending on context.

References

Alemdar, H. and Ersoy, C. (2010). Wireless Sensor Networks for Healthcare: A survey, *Computer Networks* **54**(15): 2688–2710.

Atzori, L., Iera, A. and Morabito, G. (2010). The Internet of Things: A survey, *Computer Networks* **54**(15): 2787–2805.

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**(4): 345–367.

Baldwin, C. and Woodard, J. (2008). The Architecture of platforms: A unified view. Working Paper, Harvard Business School [www document] http://www.hbs.edu/faculty/ Publication%20Files/09-034_149607b7-2b95-4316-b4b6-1df66dd34e83.pdf (accessed 9 May 2016).

Bannister, F. (2001). Dismantling the Silos: Extracting new value from IT investments in public administration, *Information Systems Journal* **11**(1): 65–84.

Bergvall-Kåreborn, B. and Howcroft, D. (2011). Mobile Applications Development on Apple and Google Platforms, *Communications of the Association for Information Systems* **29**, Article 30.

Bhaskar, R.A. ([1975] 1997). *A Realist Theory of Science*, London: Verso.

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* **29**, Article 23.

Bygstad, B. and Hanseth, O. (2015). From IT Silo Systems to Generative Infrastructures. A Study in eHealth Complexity, in Proceedings of European Conference of Information Systems (ECIS) (AIS, Münster).

Bygstad, B. and Lanestedt, G. (2014). Policies and Practices in Welfare Technologies. A Comparative Study of Norway and Japan, in Proceedings of NOKOBIT (Tapir: Fredrikstad, Norway).

Campbell, J. and Pedersen, O.K. (2006). Knowledge Regimes and Comparative Political Economy, in D. Béland and R.H. 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, Manuel (1998). Meshworks, Hierarchies and Interfaces, in J. Beckman (ed.) *The Virtual Dimension. Architecture, Representation and Crash Culture*, Princeton, NJ: Princeton Architectural Press, pp. 274–285.

DeLone, W.H. and McLean, E.R. (2003). The DeLone and McLean Model of Information System Success: A ten year update, *Journal of Management Information Systems* **10**(4): 9–30.

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 [www document] http://eprints.lse.ac.uk/47436/ (accessed 9 May 2016).

Elder-Vass, D. (2007). A Method for Social Ontology: Iterating ontology and social research, *Journal of Critical Realism* **6**(2): 226–249.

Erl, T., Gee, C., Kress, C.J., Maier, B., Normann, H., Raj, P., Shuster, L., Trops, B., Utschig-Utschig, B.C., Wik, P. and Winterberg, T. (2015). Next Generation SOA. A Concise Introduction to Service Technology & Service-Orientation, Prentice Hall.

European Commission (2011). European countries on their journey towards national eHealth infrastructures [www document] http://www.ehealthnews.eu/images/stories/pdf/ehstrategies_final_report.pdf (accessed 9 May 2016).

Gartner (2014). Bimodal IT: How to be digitally agile without making a mess [www document] https://www.gartner.com/doc/2798217/bimodal-it-digitally-agile-making (accessed 9 May 2016).

Gerring, J. (2007). *Case Study Research*, New York: Cambridge University Press..

Glaser, B. and Strauss, A. (1999). *The Discovery of Grounded Theory: Strategies for qualitative research*, New York: Aldine de Gruyter.

Ghazawneh, A. and Henfridsson, O. (2015). A Paradigmatic Analysis of Digital Application Marketplaces, *Journal of Information Technology* **30**(3): 198–208.

Greenhalgh, T., Hinder, S., Stramer, K., Bratan, T. and Russell, J. (2010). Adoption, Non-Adoption and Abandonment of an Internet-Accessible Personal Health Organiser: Case Study of Health Space, *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 Bygstad, B. (2015). Flexible Generification. ICT Standardization Strategies and Service Innovation in Health Care, *European Journal of Information Systems* **24**(6): 645–663.

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 Clark, K.B. (1990). Architectural Innovation: The reconfiguration of existing product technologies and the failure of established firms, *Administrative Science Quarterly* **35**(1): 9–30.

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

Hertzum, M. and Simonsen, J. (2010). Effects-Driven IT Development: An Instrument for Supporting Sustained Participatory Design, in Proceedings of the ACM 11th Biennial Participatory Design Conference, 61–70.

Howard-Grenville, J.A. and Carlile, P.R. (2006). The Incompatibility of Knowledge Regimes: Consequences of the material world for cross-domain work, *European Journal of Information Systems* **15**: 473–485.

Hylving, L., Henfridsson, O. and 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.

Iyer, B. and Davenport, T.H. (2008). Reverse Engineering. Googles Innovation Machine. *Harvard Business Review* April: 59–68.

Lane, D.A. (2011). Complexity and Innovation Dynamics, in C. Antonelli (ed.) *Handbook on the Economic Complexity of Technological Change*, Cheltenham, UK: Edward Elgar Publishing.

Lacity, M. and Willcocks, L. (2015). Robotic Process Automation at Telefonica O2, *MIS Quarterly Executive* **15**(1): 21–35.

Leonardi, P. (2011). When Flexible Routines Meet Flexible Technologies: Affordance, Constraint, and the Imbrication of Human and Material Agencies, *Management Information Systems Quarterly* **35**(1): 147–167.

Mccarthy, D., Mueller, K. and Wrenn, J. (2009). Kaiser permanente: Bridging the quality divide with integrated practice, Group Accountability, and Health Information Technology [www document] http://www.commonwealthfund.org/publications/case-studies/2009/jun/kaiser-permanente (accessed 9 May 2016).

Miorandi, D., Sicari, S., De Pellegrini, F. and Chlamtac, I. (2012). Internet of Things: vision, applications and research challenges, *Ad Hoc Networks* **10**(7): 1497–1516.

Munro, D. (2014). Apple Gives Epic And Mayo Bear Hug With HealthKit. Forbes. [www document] http://www.forbes.com/sites/danmunro/2014/06/03/apple-gives-epic-and-mayo-bear-hug-with-healthkit/#6ec7dc9c4167.

Niehaves, B., Köffer, S. and Ortbach, K. (2012). IT Consumerization – A Theory and Practice Review, in 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): 1053–1058.

Pawson, R. and Tilley, N. (1997). *Realistic Evaluation*, London: Sage.

Phelan, S.E. (2003). What is Complexity Science Really? *Emergence* **3**(1): 120–136.

Pollock, N., Williams, R. and D'Adderio, L. (2007). Global Software and its Provenance: Generification work in the production of organizational software packages, *Social Studies of Science* **37**(2): 254–280.

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*, Boston: Harvard Business School Press.

Sauer, C. and Willcocks, L.P. (2007). Unreasonable Expectations – NHS IT, Greek choruses and the games institutions play around mega-programmes, *Journal of Information Technology* **22**(3): 195–201.

Sommerville, I. (2010). *Software Engineering*, 9th edn, Harlow, UK: Pearson.

Sommerville, I., Cliff, D., Calinescu, R., Keen, J., Kelly, T., Kwiatkowska, M.Z., McDermid, J.A. and Page, R. (2012). Large-Scale Complex IT Systems, *Communications of the ACM* **55**(7): 71–77.

Sherer, S.A. (2014). Patients Are Not Simply Health IT Users or Consumers: The case for 'e healthicant' applications, *Communications of the Association for Information Systems* **34**, Article 17.

Sørensen, C., de Reuver, M. and Basole, R.C. (2015). Mobile Platforms and Ecosystems, *Journal of Information Technology* **30**(3): 195–197.

Smith, J.E.H. (2011). *Divine Machines: Leibniz and the sciences of life*, Princeton, New Jersey, US: Princeton University Press.

Tiwana, A. (2014). *Platform Ecosystems. Aligning Architecture, Governance and Strategy*, Waltham, MA, USA: Morgan Kaufmann.

Weick, K.E. (1976). Educational Organizations as Loosely Coupled Systems, *Administrative Science Quarterly* **21**(1): 1–19.

Willcocks, L., Lacity, M. and Craig, A. (2015). The IT function and robotic process automation paper 15/05 [www document] http://eprints.lse.ac.uk/64519/1/OUWRPS_15_05_published.pdf (accessed 9 May 2016).

Wynn, D. and Williams, C.K. (2012). Principles for Conducting Critical Realist Case Study Research in Information Systems, *MIS Quarterly* **36**(3): 787–810.

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.

About the author

Bendik Bygstad is a Sociologist who is currently a Professor at Department of Informatics, University of Oslo, and Adjunct Professor at the Norwegian School of Economics. His main research interests are IT-based service innovation and the relationship of IS and organizational change. He is also interested in IS research methods, in particular the methodological implications of critical realism. His preferred empirical approach is longitudinal process research, and he has published articles in *MIS Quarterly*, *Information Systems Journal*, *Journal of Information Technology*, *Information & Organization and European Journal of Information Systems.*

APPENDIX

Table A1 Interviewees

Interview #	*Case*	*Title of interviewee*	*Role in case*	*Date*
1	*Integration Factory*	Director	Responsible for IF	October 2014
2		Programmer	Technical integration	November 2014
3		Programmer	Technical integration	March 2015
4		Architect	Design	March 2014
5		Project manager	Customer of IF	Sept 2014
6		Sub-project manager	Requirements to IF	October 2014
7		Doctor	Requirements to IF	October 2014
8		Nurse executive	User requirements	August 2014
9		Lab specialist	Requirements to IF	August 2014
10		Doctor	User of solution	Sept 2014
12		Programme director	Plan and budgets	February 2015
13		Programme architect	Architecture informant	January 2014
14		Project manager	Plans	February 2014
15		Sub-project manager	Integration solution	March 2014
16		Lab specialist	User requirements	March 2014
17		Executice officer, Ministry of Health	Policy informant	February 2013
18		Executice officer, Directorate of Health	Policy informant	March 2015
19	*St.Hanshaugen*	Manager	Owner	January 2014
20		Nurse manager	Head of operations	January 2014
201		Nurse	User	January 2014
22		Assistant	User	January 2014
23		Patient	User	January 2014
24		Municipality director	Policy informant	October 2015
25		Vendor integrator	Vendor Dignio	February 2014
26		Vendor CEO	Vendor Dignio	February 2014
27	*Køge Hospital*	Manager	Owner	April 2015
28		Doctor	User	April 2015
29		Doctor	User	April 2015
30		Nurse	User	April 2015
31		Nurse	User	April 2015
32		Vendor CEO	Vendor Imatis	Sept 2013
33		Vendor integrator	Vendor Imatis	April 2015
34		Vendor salesrepr.	Vendor Imatis	Sept 2015
35		Platform vendor	Vendor Microsoft	April 2015
36	*Kasama City*	Executive officer, Ministry of Health	Policy informant	April 2014
37		Executive officer, Ministry of Communication	Policy informant	April 2014
38		Nurse manager	Head of operations	April 2014
39		Nurse	User	April 2014
40		Vendor executive	Vendor Hitachi	April 2014
41		Vendor architect	Developer Hitachi	April 2014
42		Vendor engineer	Developer Hitachi	April 2014
43		Vendor engineer	Developer Hitachi	April 2014