

INVERSE INFRASTRUCTURES: COORDINATION IN SELF-ORGANIZING SYSTEMS

T.M. Egyedi, J.L.M. Vrancken and J. Ubacht

DELFT UNIVERSITY OF TECHNOLOGY, DELFT, THE NETHERLANDS¹

The paper argues that a new category of infrastructures is emerging, user-driven, self-organizing and with de-centralized control: Inverse Infrastructures (IIs). IIs are not well-understood. Moreover, they represent a paradigm shift in infrastructure development. Their bottom-up development shows tension with the current socio-institutional framework for infrastructures. Internationally infrastructure laws and policies are based on a top-down and centralized view of infrastructures. Regulation is based on a control paradigm that does not fit the characteristics of inverse infrastructures and has no ways to deal with them. Policy (re)design is needed in the face of inverse infrastructure emergence.

The paper analyses how these self-organizing infrastructures emerge and develop by focusing on coordination issues. Coordination is central to self-organization. Two clusters of II cases are analyzed on what triggers coordination and how subsequent coordination takes place (coordination mechanisms). Theoretical concepts are drawn from standardization theory, from studies on Open Source Software communities, and from theories of self-organizing systems (i.e. Complex Adaptive Systems and System-of-Systems theory). The two clusters of II cases are peer-to-peer networks (e.g. Napster, Gnutella and Joost) and wireless networks (Wireless Leiden and FON).

The paper concludes that, similar to the behavior of ant colonies, II emergence can be understood as an accumulation of local attempts to optimize a situation. Complex citizen and citizen-company partnerships evolve which compete with existing infrastructure provisions and touch on public values (e.g. privacy, copyright). A policy response is needed.

There is a widespread fascination among researchers of various disciplines with order that stems from chaos, with method in madness, with complexity that results from simplicity (e.g. ant colony behavior and Internet; Holland, 1995), and with the coherent, highly valued achievements that stem from self-organization² (Open Source Software; Perens, 1999). These examples fascinate us foremost because the outcome of self-organization would lead us to suspect that a large amount of centralized orchestration takes place, whereas the opposite seems to be the case.

In the field of ICT infrastructures, a similar phenomenon is occurring. Where previously the grand design, a top-down and centralized approach, seemed imperative for harmonized infrastructure development, at present a different phenomenon is gaining ground: Inverse Infrastructures (Vree, 2003). Inverse Infrastructures (IIs) develop bottom-up, and are driven by users. Self-organization is a key element in their emergence.

In the following we will argue that IIs represent a paradigm shift in infrastructure development, a shift which policy makers have difficulties to deal with. Moreover, policy makers have little knowledge and experience

¹ We sincerely thank our two colleagues of the ICT section Jan van den Berg en Semir Daskapan for the interesting discussions and feedback they have given us, and the anonymous reviewers of SIIT2007 for their good comments. This work was supported in part by the *Next Generation Infrastructures* foundation (www.nginfra.nl).

² Self-organization refers to ‘unsupervised learning’. In the case of networks this means that there are inputs and outputs but “no feedback from the environment to say what those outputs should be or whether they are correct. The network must discover for itself patterns, features, regularities, correlations or categories in the input data and code for them in the output. The units and connections must thus display some degree of *self-organization*” (Hertz et al., 1991, p.197)

to go on because the II phenomenon is relatively new. Knowledge regarding what triggers self-organization and how self-organization can lead to seemingly highly coordinated outcomes is dispersed and limited to specific fields of science. In this paper we apply concepts and insights from self-organizing systems theory, studies of Open Source Software communities, and standardization theory³ to shed light on II-emergence. We focus on *what triggers self-organization and which coordination mechanisms underlie self-organization* in the inverse infrastructures of peer-to-peer (P2P) networks and wireless (WiFi) networks. Insight in these aspects of II-development is required to determine subsequently how the current friction between infrastructure policy and II practice can be addressed.

Before illustrating the problems the current friction may lead to, we indicate why IIs are a new phenomenon, and in what way they differ from the early development stages of other infrastructures.

Positioning Inverse Infrastructures

Early infrastructures usually started out as local and distributed initiatives. For example, where electricity is concerned local initiatives were typically taken by companies, cooperatives and municipalities (Kaijser, 1995; Weijnen et al., 2004). The same initially local and distributed pattern can be seen where gas is concerned. For example, in the 1840s and 1850s in the Nordic countries private gasworks, often built by foreign companies, served specific institutions, establishments and factories (Hyldtoft, 1995, p.79).⁴ The more regional these utilities became, the more likely the regional and national authorities would take over the production and distribution of such infrastructure services. (Kaijser, 1995) The same pattern can be observed in the development of other infrastructures (e.g. the railways and the telegraph system). Government would step in to build an integrated infrastructure and ensure universal access, reliability and quality of service (Weijnen et al., 2004).

From the mid-20th century onward until the 1990s many of our current infrastructure regimes (e.g. electricity, telecommunications, railways) evolved based on a top-down and centralized control paradigm. This paradigm was incorporated in many standards. For example, in telecommunication networks, circuit-switching and ISDN both embedded centralized control (Mansell & Hawkins, 1992).

But since the late 1990s, bottom-up initiated, decentralized and self-organizing infrastructures seem to be gaining ground. The examples range from Internet to distributed utility supply (e.g. micro co-generation of heat and power, local biogas production from waste, local waste water treatment and membrane technology for local water purification; Weijnen et al., 2004).

GOVERNANCE	ACTOR	DEFINED LOCUS	TYPE OF DEVELOPM.	CONTROL ⁵	TYPE OF CHANGE
Local Regional	Company Cooperation Local authority Regional authority	Yes	Top-down	Central	Static
Interregional National	National government	Yes	Top-down	Central	Static
International ⁶ Multi-national	Governments Company	Yes	Top-down	Central	Static
Inverse infrastructure (interconnected individuals)	Citizens, users	No	Bottom-up	De-central	Dynamic

Table 1: *Type of infrastructure governance over time*⁷.

The development of these infrastructures differs radically from earlier developments (see Table 1). Table 1 sketches the historical stages of infrastructure development from local, to regional, national, international and multi-national infrastructures, and to II. As the table indicates, II governance differs in many respects from earlier types of

³ Because self-organization is a specific type of coordination, theory that sheds light on coordination mechanisms is highly relevant - as we will see, also where self-organization is concerned.

⁴ In the Netherlands a similar pattern existed (Kaijser, 1999).

⁵ Management control; see the discussion about SoS (Maier, 1999) further on.

⁶ Based on the interconnection of national systems.

⁷ The table includes elements from Kaijser (1995) and Maier (1999).

governance – also from early infrastructure governance. New elements are: the active involvement of users and citizens in building the infrastructure; the dispersed and not predetermined locus of the infrastructure nodes (e.g. servers); the bottom-up nature of II emergence; de-centralized infrastructure control and the more dynamic nature of infrastructure change (i.e. less predictable).

We follow Vree (2003), who refers to such networks as *inverse* infrastructures (II) to emphasize that investments do not occur top-down but bottom-up. IIs have the following properties (Vree, 2003):

- bottom-up investments of users
- small heterogeneous networks coupled to larger networks
- use of existing ICT for other, unforeseen aims
- user- and consumer-driven development of infrastructures; they themselves define ‘the public interest’
- self-organizing, self-configuring, changing network development at the component and subsystem level.

In addition, as we will illustrate, technical innovation in IIs is more likely to be done by users and to occur during the diffusion phase of the infrastructure (i.e. *innofusion*; Fleck, 1988) than is traditionally the case. Table 2 summarizes the properties of II emergence and of traditional infrastructure design.

CHARACTERISTICS	DESIGN VISION	INVERSE VISION
Policy		
Initiating actors	Providers/ large companies; provider-driven developments	Users, companies, etc.; user-driven developments
Common interest defined by	Government/providers (centralization)	Users/consumers (‘empowerment’)
Investments	Top-down	Bottom-up
Technology		
Magnitude	Large-scale	Small-scale (local)
Homogeneity	Homogeneous	Heterogeneous, linked
Identifiability	Localized, predictable	Partly self-organizing, changing, unpredictable
Development		
Technical innovation	<i>Classic innovation</i> : professional, R&D-driven innovation	<i>Innofusion</i> : user innovation during diffusion

Table 2: Characteristics of the paradigm shift from designed to inverse infrastructures.

Tensions. IIs pose different requirements policy-wise (Vree, 2003, p.11). The rise of IIs raises tensions between what Perez calls ‘the three spheres of change’ (Perez, 2002, p. 156). She studied changes in techno-economic paradigms during technological revolutions. See Figure 1. Technological changes lead to institutional changes as well as economic change. More specifically,

“Because of its potential to upset or remake previously accepted relations and practices (...) the development of a new infrastructure may include a good deal of what economists have labeled ‘creative destruction’, as practices, organizations, norms, expectations, and individual biographies and career trajectories bend – *or don’t* – to accommodate, take advantage of, and in some cases simply survive the new possibilities and challenges posed by the infrastructure.” (Edwards et al., 2007, p.24)

Current infrastructure policies are based on large scale centralized production units (Weijnen et al., 2004), and do not readily accommodate the paradigm shift towards interconnected small-scale and user-driven network development. For example, where citizen-initiated city-wide wireless network coverage develops, the public/private status of such a network is unclear. Does e.g. privacy regulation which applies to public networks also apply to these IIs? Can these networks e.g. be tapped in the light of national security? Moreover, they compete with commercial networks, which have to comply with all types of requirements. Do IIs undermine the market? And finally, everyone can install a camera and connect to P2P video networks. Which privacy rules apply here – if at all?

In such cases, Perez argues, “[t]he design of appropriate policies ...requires identifying the direction of change by understanding the paradigm” (Perez, 2002, p. 164). For IIs, understanding coordination in self-organizing systems is paramount for infrastructure policy (re)design. In this paper we focus in particular on what triggers coordination and which implicit and explicit coordination mechanisms allow II emergence.

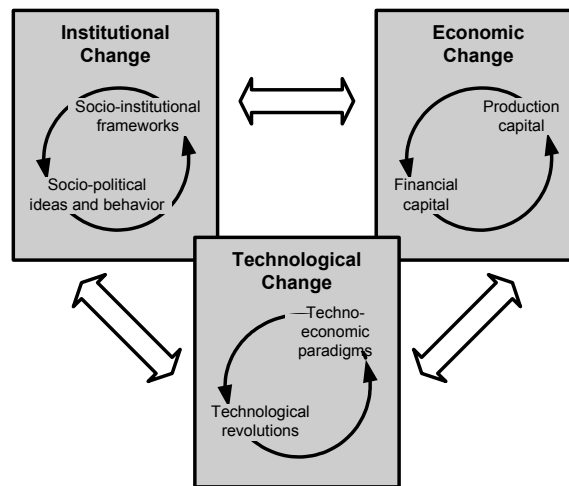


Figure 1: *Three spheres of change in constant reciprocal action (Perez, 2002, p. 156).*

Coordination in self-organizing systems

Three bodies of literature have been scrutinized for concepts and examples that throw light on how coordination takes place in self-organizing systems (i.e. coordination mechanisms, in short), i.e. :

- Standardization literature, because standardization is often equated with coordination;
- Studies of Open Source Software (OSS) communities. For OSS communities the same question applies as we pose in this paper: how do these bottom-up, self-organizing communities manage to coordinate their work?
- Theories on self-organizing systems, in particular Complex Adaptive Systems (CAS) and Systems of Systems (SoS), because these systems have strong similarities with inverse infrastructures.

A list of the relevant concepts drawn from these three bodies of literature is provided in Table 3 and **Table 4**. We discuss them in the following.

Coordination in Standardization

Classic economic studies of standardization distinguish between two types of coordination (Katz & Shapiro, 1985; Farrell & Saloner, 1988): committee standardization and coordination by the market. Committee standardization is an explicit means of coordination. It is a response to technical and actor complexity (Schmidt & Werle, 1992, 1998). The need for coordination follows from the technical interrelatedness of components and products, that is, from the “network character of many technical artefacts.” (1992, p.2) The more complex the system is in terms of number of different components, the higher and more diverse the number of actors involved. The more actor interdependencies, the higher the need to coordinate market strategies and to define common technical specifications for product and technology development. Committee standards are such specifications. They serve to create technical compatibility among products and services and direct the actions and orientation of other market players. Standards “help coordinating the actions of those who design, produce or use technical components as parts of more comprehensive technical systems” (Schmidt and Werle, 1992, p.6). Coordination of this kind is visible in several phases of infrastructure development. Standards can be used as a starting point and pre-condition for initial infrastructure development – as in inverse infrastructures; and they can function as a means to consolidate the integration of large scale infrastructures (Edwards *et al*, 2007, p.7)

In contrast, coordination by the market refers to standards that arise from a dominant market share (*de facto* standards). As we will see, for inverse infrastructures the difference between committee and ‘market’ standards is not very important because *de facto* standards can also start off inverse processes. Market coordination is, however, of interest to our analysis because it sets the scene for the concepts of “bandwagon mechanism” and “positive network externalities”. Farrell and Saloner (1988, p. 236) define the bandwagon mechanism as follows: “If an important agent makes a unilateral public commitment to one standard, others then know that if they follow that lead, they will be compatible at least with the first mover, and plausibly also with later movers.” The concept captures a leadership-induced response by the marketplace in situations where there is technical choice. It constitutes one of the most elementary mechanisms of coordination, and one which is very relevant to II development.

The notion of ‘network externalities’ is equally relevant for understanding II development – namely as a possible trigger for coordinative action. Katz & Shapiro (1985, p. 424) circumscribe it as follows: “There are many products for which the utility that a user derives from consumption of the good increases with the number of other agents consuming the good”. For example, the number of connected households increases the value of a connection to a telephone network. As we will see, the promise of increased value that results if participants coordinate their actions is a main driver of II.

In sum, several coordination-related insights from standards literature throw light on bottom-up II development, i.e.: committee standards provide a common basis for II development; the bandwagon mechanism is one of the ways in which bottom-up coordination takes place; and network externalities - and economies of scale, we might add – are possible rationales for coordinative action.

Coordination in Open Source Software (OSS) development

An important source of insight into bottom-up initiatives and self-organization is the way work is coordinated in Open Source Software (OSS) communities. Overall the individual members of these communities work in dispersed environments and without explicit orchestration. Because they use and expand the same source code one would expect a multitude of incompatible software versions to emerge (“forking”). Why doesn’t this happen to a greater degree? Which forms of implicit and explicit coordination create convergence in software development? This question has been addressed in a series of articles by Egyedi and Van Wendel de Joode. They studied three OSS communities in detail, namely the Java, Linux, and Apache communities (Egyedi, 2000; Egyedi & Van Wendel de Joode, 2004; Van Wendel de Joode & Egyedi, 2005a, 2005b, and 2006). Next to committee standardization and market coordination, they identify three main categories of coordination mechanisms that explain the degree of coordinative outcomes in OSS development: regulatory coordination, operational coordination, and coordination by authority. These are listed and exemplified in Table 3.

The coordinative nature of most examples will speak for itself. For some examples this may not be clear. We therefore briefly explain their coordinating workings. I.e. regarding

- to-do lists, OSS developers sometimes draft common to-do lists, which signal to third developers what issues other members of the community find important;
- orphanages, lists are drafted that contain the projects that have been abandoned and are in need of a new maintainer. These lists have a coordinative effect, since the likelihood that existing projects are continued increases and the likelihood of creating rival software decreases;⁸
- concurrent versioning systems (CVS), these enable multiple developers to work on the same version of the source code simultaneously.⁹ The software can be downloaded, commented upon, changed, and modifications can be tested or even removed, etc. The CVS can be said to increase coordination because those who disagree with the modifications are challenged to prove otherwise - within the confinement of the community. Without the option of “in-house divergence”, dissidents would be forced to prove their point outside the community, which could lead to competing software;
- coordination by authority, this occurs “if one player is nominated in advance to choose [among alternatives], and others then defer to him” (Farrell & Saloner, 1988). Their circumscription emphasizes formal authority. We stretch their circumscription to include coordination based on informal authority. Authority can be based on a variety of resources: technical expertise, economic success, etc.

Reviewing Table 3, some mechanisms are obviously more forceful and explicit than others, such as the use of certain IPR licenses and the leadership which the Linux initiator Linus Torvalds can exert in Linux development. They impose coordination (*coordination push*). Others are more implicit and entice rather than impose coordination (*coordination pull*), such as the unifying effect of reading the same instructions, or the use of counters on the community website; these indicate the number of downloads of a specific software version which triggers the bandwagon mechanism. In the case of II development, its self-organizing, bottom-up nature indicates that we are more likely to find examples of coordination pull (i.e. enticement) than coordination push (i.e. top-down, imposed coordination).

The examples further show that the coordinative moment in OSS development can be pinpointed at the start of a new software development trajectory (input coordination, e.g. starting out with the same code), during the development (process coordination, e.g. CVS), and at the end of the - usually intermediate -trajectory (output coordination; selectively assigning a compatibility logo to software). Building on the previous section, the role of standards in initial II development is that of input coordination in OSS development.

⁸ <http://www.debian.org/devel/wpp/orphaned> (March 2005)

⁹ www.cvshome.org, August 2002

All three coordinative moments are relevant for analyzing coordination in II development. Because our cases focus on *early* II development, for us the first two coordinative moments are most relevant.

COORDINATION MECHANISMS	EXAMPLES
Committee standardization	Market and technology coordination in committees <ul style="list-style-type: none"> • Founding a standards community and standards procedures (e.g. Java) • Standardization initiatives
Market coordination	Coordination in the market place <ul style="list-style-type: none"> • Individual's reputation as an expert • Software distributions (i.e. distribution of a select software version of e.g. Linux) • Desirability of a trademark (e.g. Java Compatibility logo) • Consumer expectations • Market share as a sign of product quality • Project activity on website, high level of activity may indicate technical excellence
Regulatory coordination	Company rules and government regulation enforce coordinative behavior <ul style="list-style-type: none"> • Participation agreements (e.g. OSS community membership registration) • Intellectual Property Right licenses (e.g. Apache Software License, Linux GPL) • Contracts (<i>Technology License and Distribution Agreement</i> for Java) • Trademarks (e.g. Java-Compatible logo)
Operational coordination	Developers in open source communities use several tools to support their activities. These tools coordinate, i.e. focus and structure, their work. <ul style="list-style-type: none"> • software support tools, these support the development of interoperable programs • instructional books • certified training programs • manuals (e.g. the policy manual of the Debian community¹⁰ indicates how packages are to be assembled.) • test suites • reference implementations • concurrent versioning systems (CVS)/ Subversion: enables software developers to work on the same version simultaneously • to-do lists (ideas are parked there to be taken up by others) • orphanages (for OSS projects from which the project leader or maintainer withdraws and which is open for adoption)
Coordination by authority	<ul style="list-style-type: none"> • gatekeepers, informal or formal hierarchy.

Table 3: *Coordination mechanisms in open source software development.*

Theories on Self-Organizing Systems

Complex Adaptive Systems (CAS) and Systems of Systems (SoS) have a lot in common with inverse infrastructures. In these systems, too, forms of coordination take place in sometimes unexpected ways. We will therefore examine these theories systems on insights relevant to II development.

Complex Adaptive Systems (CAS).

Network complexity can stem from very different sources (Strogatz, 2001, 268-269). For example, complexity may emerge from the sheer number of network components (Gell-Mann, 1994). The complexity in 'CAS' arises because the component parts can adapt and evolve based on the history of their interactions with other parts of the system and/or with their environment (McIndoe's, 2005, p.5). In CAS control is dispersed and decentralized. De-centrally attempts are made to optimize the situation at a local level. This then leads to emergent system behavior – as opposed to pre-determined – and lies at the roots of self-organization (Gell-Mann, 1994; Holland, 1996; Dooley, 1996)

¹⁰ www.debian.org/doc/manuals/project-history/ch-intro.html.s1.1, May 21, 2003

These types of systems have their analogies in the animal world. For example, Kelly (1994) speaks of “swarm systems” and “hive mind”. The classic example of a CAS is the behavior of an ant colony when it is looking for food (Wheeler, 1928). How do ants communicate and coordinate their actions? CAS researchers have formulated ant algorithms to explain their group behavior. Individual ants act within a bounded range of behavior. Coordination is not designed in, but results from cumulative responses of individual ants to a change in their environment. When food is found, the increasing intensity of the pheromone trace, a substance secreted when moving, indicates to other ants that food is near. In this manner the quickest pheromone track to food becomes the most attractive one – a process which is comparable to the bandwagon mechanism.

In the case of ants simple and uniform behavior coupled with reactions to traces of pheromone together create the coordinative complexity necessary for the group to survive. Likewise the flocking behavior of bats can be simulated by three simple rules: don’t bump, keep up with neighbor, and don’t stray (Kelly, 1994, p.11). The examples indicate that initial simplicity¹¹ can lead to complexity.

System-of-systems (SoS)

The term “system-of-systems, as commonly used, suggests assemblages of components that are themselves significantly complex, enough so that they may be regarded as systems and that are assembled into a larger system” (Maier, 1999, p.269). Some SoS authors emphasize complexity, some the geographically distributed nature of the subsystems, others the principle independence of the subsystems and the evolutionary nature of the overall system. To Maier, whose definition we center on, a system-of-systems is “an assemblage of components which individually may be regarded as systems, and which possess two additional properties:

- Operational independence of the components: If the system-of-systems is disassembled into its component systems the component systems must be able to usefully operate independently. (...)
- Managerial independence of the components: The component systems not only can operate independently, they do operate independently. The component systems are (...) integrated but maintain a continuing operational existence independent of the system-of-systems.” (Maier, 1999, p.269)

These two properties also apply to IIs. The components are operationally and managerially independent. They work in a collaborative way, and do so voluntarily, which means that system control is decentralized.

Maier’s definition has implications for SoS design and provides insight in the coordination properties of interfaces. SoS design requires greater emphasis on interface design than is usual. Interfaces are the greatest leverage, but also the greatest danger. Since “components should be severable from the system-of-systems without destroying the desired emergent behaviors” (p.272), firstly, interfaces between components become critically important in, for example,

- how they foster or discourage collaborative emergent functions in the system-of-systems (p.279) and
- how they support a) vertical severability (i.e. the ability to remove or add a physical component to the SoS, and b) horizontal severability (i.e. ability to remove or add applications or functions to the SoS independently of physical components).

He holds that in SoS the design of the interface standards, standards of communication included, is more important than the design of the components. The interfaces are the architecture (p.280), and vice versa (p.273).

Secondly, “[i]f a system requires voluntary collaboration, the mechanism and incentives for that collaboration must be designed in.” (p.273) Maier illustrates this, firstly, with the Internet servers, where the cost of collaboration is relatively low and the benefits high; and secondly, with “a situation in which each participant’s well-being is partly dependent on the well-being of the other participants” (i.e. the joint utility approach, p.274).

Maier’s emphasis on the importance of interfaces in SoS aptly describes the crucial nature of standards as a starting point for II development. They need to at least allow – or even provide incentives for – bottom-up collaboration.

Lastly, Maier distinguishes categories of SoS by whether they have a central purpose and whether they have central control. For example, when the Internet was still part of the DARPA project, there was a central purpose and central management control (i.e. *directed* SoS). At present, the purpose of Internet still exists, but the system depends on voluntary cooperation of service providers, a SoS which Maier refers to as a *collaborative* SoS. His argument focuses specifically on such SoS. The SoS which has neither central control nor a central purpose is the *virtual* SoS (p.178). In their pure form IIs are virtual SoS, with no central control nor a central purpose. The cases will illustrate, however, that in practice IIs may vary on both dimensions.

¹¹ Uniformity is not needed. E.g. in the case of bees looking for a new place to settle, there is no uniformity at the outset but still a CAS-like phenomenon; i.e. there are different bee-roles.

Summary of coordination issues

In the above we discussed coordination issues in three strands of literature (i.e. standardization, OSS, and self-organizing systems). They are listed in **Table 4**— with a reference to **Table 3** for completion. To which incentives for coordination and coordination mechanisms in II development do they sensitize us? With these insights in mind the II cases in the next section will be presented and analyzed. We will also examine these inverse infrastructures on additional examples of coordination.

THEORIES	COORDINATION ISSUES
Standardization	Committee standards (common basis and guide posts) Bandwagon mechanism Network externalities and economies of scale.
Open source communities	Coordination push - coordination pull Coordinative moments: input, process, output coordination (see Table 3 for an elaborate list of OSS coordination mechanisms)
CAS	Local attempts to optimize situation lead to emergent systems Cumulative responses of individual ants to change in environment
SoS	‘Designed in’ collaboration Standard interfaces needed for operationally and managerially independent components Trigger: if cost of collaboration is low and benefits high Trigger: joint utility approach: when participant’s well-being depends on the well-being of others Coordination dimensions: central purpose and central control

Table 4: *Coordination Issues.*

Case studies

In the following two cases of inverse infrastructures in the field of ICT are analyzed, i.e. peer-to-peer networks and wireless networks. With help of the sensitizing concepts derived from literature they are analyzed on what triggered their self-organization and which mechanisms of coordination helped to develop them.

P2P networks

A P2P network (Oram, 2001; Parameswaran et al, 2007) is an inverse infrastructure on top of the Internet infrastructure. P2P networks are mainly used for file exchange and are currently among the most important sources of data traffic on the Internet¹². A P2P network consists of a number of servers, the peers, that in principle all play an equal or similar role. Therein it contrasts with a client/server networks (Taylor, 2004). in which clients and servers have essentially different roles (clients are allowed to take the initiative in communicating with a server and not the other way around). P2P networks have a long history, dating back to the beginning of the Internet. The Usenet newsgroups can be considered as one of the first, and still very primitive, P2P networks. The purpose was and is to exchange opinions by posting messages to a newsgroup. Exchanging files, attached to the messages, was a secondary functionality. Newsgroups still exist and are heavily used for file exchange, notwithstanding their two main drawbacks, i.e. of limited capacity of the news servers and the primitive ways in which they allow searching for content. These two problems were considerably improved in the next version of P2P networks, of which Napster is the best known example (Parameswaran *et al*, 2007). In essence Napster centralized content indexing, which means that one could search the Napster database for specific content and then be referred to one of the peers offering that specific content. By that time, P2P file sharing was already heavily used for the purpose of sharing copyrighted material (music, commercial software, movies, books, etc.). This caused a number of law suits against P2P network participants. The centralized content indexing concept of Napster turned out to be too vulnerable and the Napster website was dismantled. However, it was not hard to improve on the Napster concept, and probably the first to do so was the designer of the Gnutella network (Adar & Huberman, 2000).

In the Gnutella network, the index as well as the content was distributed among the peers. A search request could in principle travel through the whole Gnutella network of peers, until the requested item was found. Legally

¹² Presentation by Tribler Designer, Johan Pouwelse: www.st.ewi.tudelft.nl/~pouwelse/tribler_TBM_vak2007_v2.ppt.

this was a sturdy concept, but it was a bit inefficient. A host of Gnutella-like P2P networks then appeared, some of them based on intermediate approaches - between centralized and purely P2P - to indexing content, such as Kazaa (Barreau, 2003). In these networks, storage was strongly distributed, indexing had become reasonably efficient, but transport was still a problem. With growing file sizes (movies, TV-programs), this problem was felt strongly, especially in the case of new material for which there was a large demand and still few peers able to supply it. It was also a problem that one was dependent on the other peer staying on-line until downloading had finished. This problem was tackled by the development of bit torrent, an approach to P2P networking in which the transport function is also strongly distributed, leading to very fast transfers when compared with the previous generation of P2P networks. Bit torrent however was a step backwards in content indexing. It re-introduced the indexing sites, a number of which were taken down as a consequence of law suits against the site owners, only to be quickly replaced by other sites, often in countries with legislation less vulnerable to this kind of litigation.

Currently, a number of efforts are underway to improve P2P networking up to the point that they support streaming data (Joost, Tribler, Zudeo, etc.). This will make it possible for vast numbers of consumers to become producers of content. Most of these efforts are aiming at so-called P2P television. Everybody with a digital video camera can then create content and have it instantly broadcasted on the Internet.

Drivers behind P2P networks

The growth of P2P networks is based on a desire to communicate and an interest in sharing content with other people (knowledge, entertainment: books, music, etc.). P2P networks start by somebody writing a piece of software, which we will call the peer-application that makes the network possible. This piece of software is then taken up by a number of individuals, who may have spread the word in many ways (by mouth, e-mail or via an older P2P network). The incentives to create a new network are often just 'for fun', the challenge of attracting many users (which then be targeted for advertising or other commercial activities), or the prospect of fame. The users are driven by a desire for communication and content, and curiosity. In most networks users are enticed to contribute content by the tit-for-tat principle: if you want to download something you have to offer content for others to download, and at least the content that you just downloaded yourself. The tit-for-tat principle is a nice example of coordination pull. A small number of participants that upload content without getting anything in return is sufficient to start the process. The presence of these starters is usually statistically guaranteed. An important reason to upload content is, of course, that some people would like to distribute their own content.

If the network is successful, the peer-application usually undergoes updates. Most of these networks feature copyrighted content. The copyright owners try to devise ways to fight these networks, if not with law suits, then by any other way that hampers the illegal exchange of information. Such attacks are then, as far as possible, countered by peer-application updates. Up until now, the networks have turned out to be quite robust, with none of the known attacks being very successful.

Coordination in P2P networks

Coordination in P2P networks can be described from the viewpoint of the designer of the peer-application and the network participant. From the designer's point of view, the most important form of coordination occurs by using the same peer-application (input coordination) and by means of the tit-for-tat mechanism (Cohen, 2003) present in all successful peer-applications (process coordination).

Coordination among participants occurs because they look at what others are doing (bandwagon effect). For P2P networks Metcalfe's law, which says that the value of a network is proportional to the square of its size (Gilder, 2002; Briscoe *et al*, 2006), holds as much as for any other network. At stake are network externalities and the prospect of economies of scale. Users are further interested in the prospect of a large network because it promises to be future-proof. Although not all initiatives for P2P networks have been successful, the market is big enough to allow a host of networks to coexist.

Once the network is established, implicit and explicit forms of coordination take place among those who further develop the software that runs on each node and those who develop the more advanced functionalities. In this phase, multi-network applications appear that cover several P2P networks, and make it easier for the user to find and retrieve the desired content. These multi-network applications are a step towards the future merger of networks into one big P2P network, comparable to the way Internet developed.

The above description of P2P network development shows that where - coordination by - authority is at stake, usually informal types of authority and implicit sources of power will be involved. However, to a lesser degree also more formal and top-down types of coordination occur. For example, the bit torrent network features content moderators (i.e. *gatekeepers*) to increase the quality of the content offered. This example also illustrates

why pure inverse infrastructures – both decentralized control and purpose – are hard to find, since within their overall bottom-up approach a nested top-down approach may be used to ease specific coordination problems.

Wifi networks and FON

Wifi is a standard for short range wireless data communications.¹³ Its range is in the order of 10 to 100 meters, depending on equipment and circumstances, making it especially useful as a LAN (local area network) technology. It was developed in the late nineties, as an IEEE family of standards (IEEE 802.11, with versions a, b, g, n, etc., offering different bandwidth and range characteristics). If large numbers of users apply it, this may result in network coverage over larger areas, even whole cities. If in addition some of these users hook up their network to the Internet using a wired broadband connection, this may result in easy, wireless Internet access all over a city. This idea has been realized to various degrees in many cities (Tallberg, 2006). In the Netherlands, probably the best known one is the Wireless Leiden network in the city of Leyden.

Some people aim at worldwide coverage of wireless internet access, at least in populated areas. FON is such an initiative (Tallberg, 2006).¹⁴ One can participate in FON by buying a piece of hardware, a wireless router that also allows a wired connection to the internet. FON is a for-profit organization. It recognizes three membership categories to allow for different cost-benefit considerations: Alien, Linus or Bill. An Alien does not offer a FON service him- or herself, and has to pay a small fee for internet access via the FON network. A Linus (after Linus Torvalds) offers access for free, and in return is allowed to use FON access points at no cost. A Bill (after Bill Gates) offers access to aliens and receives half of the fee the aliens have to pay (i.e. in the order of 3 dollars a day).

Although FON has attained a considerable size and is present in most Western countries and larger towns, it is not clear whether FON will actually succeed in the long run. As a commercial organization it charges its users - also the Linusses pay about 40 dollars for the special FON router. As such it is up against serious competition from networks like the GSM/GPRS and UMTS wireless telephone networks, which also offer internet access and currently feature a much better geographical coverage.

Drivers behind FON. The people behind FON are clearly driven by commercial motives. Users are attracted by the cheap or free internet access it offers in populated areas – albeit with limited coverage. The price of the FON router is comparable with similar products, so users can solve a local need for internet access all over house and garden, and have the FON-service as part of the bargain. Additional services are being offered on top of the FON network, such as integrated FON-based Skype phones. Moreover, it takes a firmware update to use the FON router for other purposes (this update is not hard to find on the internet).

But competing networks and uncertainty about its future are hampering FON's deployment. Users have several alternatives based either on cell phone access to internet or other Wifi hotspots offered at many hotels and public places, such as libraries. Thus for users, network externalities are again the most important driver.

Coordination in FON. Three phases can be distinguished in the development of the FON network. The first phase consists of the development of local wireless networks based on the IEEE standard developed in the late 1990s. In this phase, the standard provides a common technical focus which is adhered to locally.

The second phase covers the emergence of metropolitan Wifi-based networks. Coordination in the Wireless Leiden project, for example, was in the hands of the organizers and volunteers and based on direct contact.

The third phase consists of the development of a world-wide Wifi-based network. To develop FON, the FON company equipped participants with the necessary hardware, the Fonera router. The router constituted the common interface needed for the FON network (*input coordination*). The company itself organized and coordinated the network's further development. It combined *coordination by authority* (e.g. defining the rules) with *coordination pull* (e.g. three membership classes to entice different people to participate).

From the participant's point of view, regular use of the FON network depends on expected network externalities (e.g. availability at the destination), bandwidth effect, and economies of scale.

Conclusion

What triggers self-organization and which coordination mechanisms underlie self-organization in the inverse infrastructure cases of peer-to-peer and wireless networks? The cases illustrate two phases in II emergence, the 'initiative' and the 'early expansion' phase. Different modes of coordination play a role in both phases. The cases

¹³ http://www.wi-fi.org/white_papers/whitepaper-010204-wifipublicaccess/ consulted 29 August 2007.

¹⁴ www.fon.com/en/, consulted 29 August 2007.

show that II emergence can be understood as an accumulation of local attempts to optimize a situation (e.g. download music, or install an in-house wireless network). Highly relevant for the 'initiative' phase is that participants have already made an initial investment to address a local need or solve a local problem. The step towards exploiting the cumulative possibilities of local investments and participating in inverse infrastructure development is then relatively small. What is needed on the designers' side is in essence agreement to use a common standard or application and the incentive to start collaborating (e.g. fun, challenge or commercial prospect), i.e.: a common base and, initially, a common focus.

Important is also that users have an initial incentive to cooperate and participate. Apart from general incentives such as a desire to communicate and exchange content, users need to be enticed to play an active role as a node in an inverse network (*coordination pull*). They must be persuaded of the relatively low cost of collaboration and the high benefits involved. Moreover coordinative incentives and collaboration can be designed into e.g. standards (e.g. TCP/IP) and software (e.g. tit-for-tat principle). The degree to which coordination is encoded and self-propelling is critical.

Coordination mechanisms are as vital during the early expansion phase as they are in the initiative phase. The bandwagon mechanism supports ongoing involvement and network expansion. They are further fed by increased network externalities.

The inverseness of IIs is characterized by bottom-up decentralized control. However, IIs do not develop in limbo. Because there is no central control, by necessity other compensatory means of coordination are required. To start with, the initiators and participants must have rough consensus about the network's purpose. Moreover, nested within the overall bottom-up approach to network development, top-down modes of coordination are used to ease specific problems (e.g. use of gatekeepers, coordination in smaller group, content moderators).

The number of company-driven, FON-like inverse infrastructures is growing. This makes the question all the more important how to deal with such complex, hybrid, potentially unstable informal partnerships between a private company and citizens which together deliver infrastructure facilities and services that were formerly labeled as public.

By focusing on coordination in self-organizing systems, this study has tried to capture the essence of what drives and initially sustains de-central infrastructure development. A better understanding of II is a pre-condition for developing infrastructure policy that can successfully address the privacy, copyright and other regulatory issues which arise from mixed public-private infrastructures. Bottom-up coordination in IIs and the degree of complexity which is involved escapes the control-oriented approaches in current governance regimes. A mind-shift is needed. What Maier (1999, p.283) holds to be true for designers is also applicable to infrastructure policy makers: It would be a "fundamental error [if a self-organizing systems was] treated (...) as if it were a monolithic system [and that policy makers] believe they have control where they do not (...)." But whether there is a role for policy to play and, if so, which role, is a matter for further study.

Further study is needed to improve our understanding of IIs as well as of the relationship between different types of infrastructures and different approaches to policy. We specifically recommend two strands of additional research. First, a core question is to which degree decentralized control in self-organizing systems is self-sustaining. For example, does increasing the scale of use of inverse infrastructures ultimately lead to centralized control? The answer will have an impact that reaches far beyond infrastructure policy.

Second, the relationship between type of infrastructure development (classic or inverse) and modes of governance (formal government intervention versus self-regulatory regimes) needs further exploration. Should and can the existing governance regime address inverse infrastructures? If so, in what manner? A review of current studies into the tension between government policy and Internet governance promises to provide useful insights.

References

- Adar, E. & B. Huberman (2000). Free Riding on Gnutella, First Monday, volume 5, number 10.
- Barreau, M. (2003). KaZaA, iMesh, eDonkey et tous les autres, CampusPress.
- Briscoe, B., Odlyzko, A., & B. Tilly (2006). Metcalfe's law is wrong, IEEE Spectrum, Vol. 43, Issue 7, pp. 34-39.
- Cohen, B. (2003). Incentives Build Robustness in Bit torrent, Workshop on Economics of Peer-to-Peer Systems, <http://www.bittorrent.org/bittorrentecon.pdf>
- Dooley, K. (1996), "Chaotic Dynamics and Autonomous Agents in Manufacturing," *Chaos Network*, 8(2): 14-23.
- Edwards, Paul N., Steven J. Jackson, Geoffrey C. Bowker, and Cory P. Knobel (2007). Understanding Infrastructure: Dynamics, Tensions, and Design (Ann Arbor: DeepBlue), <http://hdl.handle.net/2027.42/49353>.
- Egyedi, T.M. (2000). 'Compatibility Strategies in Licensing, Open Sourcing and Standardization: The case of JavaTM', In: H. Coenen, M.J. Holler & E. Niskanen (Eds.), 5th Helsinki Workshop on Standardization and

- Networks, 13-14 August 2000. Government Institute for Economic Research, Helsinki, *VATT-Discussion Papers* 243, pp. 5-34.
- Egyedi, T.M., & R. van Wendel de Joode (2004). 'Standardization and Other Coordination Mechanisms in Open Source Software'. *International Journal of IT Standards & Standardization Research*, 2/2, pp.1-17.
- Farrell, J. and G. Saloner, "Coordination Through Committees and Markets," *Rand Journal of Economics* 29 (2) (1988): 235-252.
- Fleck, J. (1988). *Innofusion or diffusion? The nature of technological development in robotics*. Edinburgh PICT Working Paper, No. 4.
- Gell-Mann, Murray (1994). *The Quark and the Jaguar: adventures in the simple and the complex*. New York: Freeman & Co.
- Gilder, G. (2002). *Telecosm, the World after Bandwidth Abundance*. New York: Free Press.
- Hertz, J., Krogh, A., & R. Palmer (1991). *Introduction to the Theory of Neural Computation*, Amsterdam: Addison-Wesley.
- Holland, J.H. (1995). *Hidden Order: How Adaptation Builds Complexity*. Reading, MA: Addison-Wesley.
- Kaijser, A. (1995). Controlling the grid: the development of high-tension power lines in the Nordic countries. In: A. Kaijser & M. Hedin (Eds.), *Nordic Energy Systems: Historical perspectives and current issues*. Canton, MA, USA: Watson Publishing International.
- Kaijser, A. (1999). 'Striking Bonanza. The Establishment of a Natural Gas Regime in the Netherlands'. In Olivier Coutard (Ed.), *Governing large technical systems*. London: Routledge, pp. 38-57
- Katz, M. and C. Shapiro, "Network Externalities, Competition and Compatibility," *American Economic Review* 75 (3) (1985): 424-440.
- Kelly, K. (1994). *Out of control*. New York: Addison Wesley.
- Lessig, L. (2006). *Code version 2.0*. New York: Perseus.
- Maier, M.W. (1999). Architecting Principles for Systems-of-Systems. *Systems Engineering*, 2:1, p.267-284.
- Mansell, R. & R. Hawkins (1992). Old Roads and New Signposts: Trade Policy Objectives in Telecommunication Standards. In: F. Klaver & P. Slaa (Eds.), *Telecommunication, New Signposts to Old Roads*. Amsterdam: IOS Press, pp.45-54.
- McGowan, D. and M.A. Lemley, "Could Java Change Everything? The Competitive Propriety of a Proprietary Standard," *Antitrust Bulletin* 43, 715 (1998)
- Oram, A. (2001). *Peer-to-Peer: Harnessing the power of disruptive technologies*. O'Reilly Media.
- Parameswaran, M., Susarla, A., Whinston, A.B (2007). P2P networking: an information sharing alternative, *IEEE Computer*, July 2007, Vol 34, Issue 7, pp. 31-38.
- Perens, B. (1999) The Open Source Definition, in *OpenSources Voices from the Open Source Revolution*, DiBona, C., S. Ockman and M. Stone (Eds) O'Reilly & Associates, Sebastopol, pp. 171-189.
- Perez, C. (2002). *Technological revolutions and financial capital: The dynamics of bubbles and golden ages*. Cheltenham UK: Edward Elgar.
- Schmidt, S.K. and R. Werle, "The Development of Compatibility Standards in Telecommunications: Conceptual Framework and Theoretical Perspective," in *New Technology at the Outset: Social Forces in the Shaping of Technological Innovations* ed. M. Dierkes and Ute Hoffman (Boulder, CO: Westview, 1992);
- Schmidt, S.K. and R. Werle, *Co-ordinating Technology: Studies in the International Standardization of Telecommunications* (Cambridge, MA: MIT Press, 1998).
- Tallberg, M.(2006). P2P-based Roaming Between Home WLAN Hotspots, Seminar on Networking Business, TKK Networking Laboratory, http://www.netlab.tkk.fi/opetus/s383042/2006/papers_pdf/F1.pdf, consulted 29 August 2007.
- Taylor, I.J. (2004). *From P2P to Web Services and Grids: Peers in a Client/Server World*. Springer.
- Vree, W.G. (2003). *Internet en Rijkswaterstaat: een ICT-infrastructuur langs water en wegen*. Inaugural speech, Delft University of Technology.
- Weijnen, M.P.C., E.F. ten Heuvelhof, P.M. Herder and M. Kuit. (2004). *Next Generation Infrastructures. Business Plan*. Delft: Next Generation Infrastructures Foundation.
- Wendel de Joode, R. van & T.M. Egyedi (2005a). 'Handling variety: The tension between adaptability and interoperability of open source software', *Computer Standards & Interfaces*, 28, pp.109-121.
- Wendel de Joode, R. van & Egyedi, T.M. (2005b). 'Handling variety in Java & Linux: Coordinating Mechanisms for the Future Generation?', in S. Bolin (Ed.), *The Standards Edge: Future Generation*. Felton, CA: Bolin Communications, pp.365-378.

- Wendel de Joode, R. van & T.M. Egyedi (2006). 'Standardization and Other Coordination Mechanisms in Open Source Software', in K. Jakobs (Ed.), *Advanced Topics in Information Technology Standards and Standardization Research*, Vol. 1, pp.71-90.
- Wheeler, W.M. (1928). *Zatapinoma*, a new genus of ants from India. *Proceedings of the New England Zoological Club*, 10, 19–23.

