Standards enhance system flexibility?
Mapping compatibility strategies onto flexibility objectives

Tineke M. Egyedi
Standards enhance system flexibility?
Mapping compatibility strategies onto flexibility objectives

Tineke M. Egyedi

*Delft University of Technology*


For comments please contact Tineke Egyedi at: Delft University of Technology; Department of ICT/ Standardization, Faculty of Technology, Policy and Management; Jaffalaan 5, PO Box 5015, 2600GA Delft, The Netherlands; +3115278 6344 (phone)/ -3741 (fax); T.M.Egyedi@TBM.TUDelft.nl.
Abstract
Large Technical Systems (LTSs) such as information networks and transport systems are difficult to change when they are entrenched and lack flexibility. The paper argues that standards enhance system flexibility. This line of argument is explored and elaborated into a conceptual framework that aims to support practices and policies of flexible LTS design. First, the behavioral scope of standards is analyzed in the context of compatibility. The paper identifies several ways of achieving compatibility (compatibility strategies). These can be portrayed by four compatibility dimensions, one of which is standardization. In combination, these dimensions raise awareness to a new range of potential compatibility strategies.

Next, since flexibility is a mean and not an aim in itself, the paper specifies a number of flexibility aims common in LTSs. As an illustration, an overview is given of those in the field of ICT.

Finally, a framework is drawn up that integrates the two sides of the argument. It lists the main flexibility objectives, and summarizes the compatibility strategies that increase system flexibility – including standardization. Each strategy has its caveats. The paper addresses those related to standards.

Keywords: system design; flexibility objectives; compatibility; standards; entrenchment; compatibility strategies; compatibility dimensions; dedicated gateways; middleware; agent technology; reusability; transparency.

Acknowledgement
The Dutch Ministry of Transport, Public Works and Water Management / Directorate-General of Public Works and Water Management funded the 2001 workshop of the European Academy for Standardization (EURAS) in Delft, on which this article builds forth. It also funded the additional research required to write this article. I gratefully acknowledge its support for both activities. I further sincerely thank the EURAS 2001 workshop participants for the highly inspiring contributions to the first experiment, and in particular Taavi Valdlo, Heide Coenen and Rudi Bekkers, Ragna Zeiss, Ole Hanseth, Timothy Schoechle, Casper van der Veer, and several anonymous participants whom I quote in this paper. Lastly, I want to thank my colleagues of the ICT department for their comments on the draft, and Wim Vree in particular for our discussion about the figures 4 and 5; the department of Innovation Studies of the Utrecht University, Faculty of Geographical Sciences, and Anke Sijtema of the Ministry of Transport, Public Works and Water Management for their addition to figure 4; Taavi Valdlo and Kai Jakobs at the EURAS 2002 workshop in Wroclaw, Poland, for their remarks. They may not agree with the paper’s content.
"The ideally flexible infrastructure would be one that was designed to evolve, itself, with emerging technologies (...)" (Duncan, 1995, p. 44)

1. Entrenchment in Large Technical Systems

Large technical systems (LTSs) such as nuclear power plants and the transportation infrastructure often seem impervious to change while change may be very desirable from a societal point of view (e.g. sustainable energy and transportation). Given the complexity of such systems and their embeddedness, it is understandable that this should be the case. LTSs consist of countless numbers of interdependent socio-technical components and subsystems. They comprise technical artifacts as well as, for example, the institutional and regulatory contexts of artifact use and production. Organizations and companies emerge that develop and sustain the system. Each specializes in certain tasks, develops technical add-ons and complementary products, gains experience with part of the system, etc. As the LTS expands, the number of and interdependencies between actors and artifacts grows. Over time, these interdependencies crystallize, solidify, and make manifest a process of socio-technical entrenchment (Collingridge, 1980, p.47).

Changes to an LTS are only possible at the cost of re-adjusting the technologies and other socio-technical arrangements that surround it. The larger the vested interests, the higher the costs and the more difficult it is to introduce changes to the system.

A well-documented example of entrenchment is that of polyvinyl chloride (PVC) production. From the early 1930s onwards, its production has posed health and environmental risks (Mulder & Knot, 2001). The danger ranged from health risks for workers and those living near the PVC production and processing plants (toxicity and carcinogenity caused by vinyl chloride; Miamata disease due to mercury emission) to the dioxin found in cow milk as a result of incineration of PVC waste in the 1980s. Despite recurrent public protest, PVC is still produced and used nowadays. Indeed, in reaction to criticism and public fear the PVC industry has optimized the production chain. Optimization, in turn, has reinforced the process of PVC entrenchment (Mulder & Knot, 2001, p.284), making the industry's conversion to non-chlorinated plastics even less likely.

To the onlooker, such large technical systems appear to have their own technological momentum (Hughes, 1983). This "(...) pushes the system along a path-dependent process of technological change (...)" (Davies, 1996, p.1148). Apparently, unless something radical happens, no noteworthy deviations from the set path will occur.

Standards play a crucial role in the evolution of LTSs (Joerges, 1988, p.30). This applies to, for example, health and safety standards as well as to the compatibility or interoperability standards focused on in this paper. By definition, these standards also play a role in the inevitable process of entrenchment. Standardization processes and the

1 I paraphrase Collingridge’s definition here. Where he emphasizes the technical, I emphasize the socio-technical character of the entrenchment process.
2 The word is used here loosely to refer to formal, consortium as well as de facto standards. It will be specified in section 4.
3 Standardization is defined in more than one. In the current context, the definition of the official international standards bodies suffices. "Standardization is an activity of establishing, with regard to actual
standards that result foremost seem to work as catalysts of entrenchment. There are two main, related reasons to think so. Firstly, standards codify existing knowledge and practices. In Reddy's wording "(...) standardization (...) is an attempt to establish what is known, consolidate what is common, and formalize what is agreed upon."(1990, p.59)\(^4\) Codification and, more in general, institutionalization processes are a main source of entrenchment.

A second source of entrenchment is the interrelatedness of multiple components which characterizes an LTS. These components are complementarities (David & Greenstein, 1990, p.7). Their relationship is often defined by standards. The set of interrelated products based on the A4 paper format illustrates this. The format provides the specifications needed to develop compatible products and assets. It determines the interfaces of all kinds of paper processing machines (e.g. copying machines, telefax's and printers) and office requisites (e.g. folders, computer software). It also eases the entry of new competitors and supplementary players in the market place, which explains why widely applied standards such as the A4 format often lie at the basis of a technological trajectory. They increase the amount of and interdependence between actors and artifacts, and further technology development (Cargill, 1989; Reddy, 1990, p.56). Stabilization is a characteristic feature of entrenchment, which in the long run cannot be avoided.

Although standards may be part of the problem of entrenchment, counter-intuitively, there are also indications (e.g. Mulgan, 1990) that standards can be part of its solution. This line of thought is explored in the following.

2. Research approach

Before doing so, however, some methodological issues are turned to. The problem definition is refined, first, by rephrasing entrenchment as a flexibility problem (section 2.1), and, second, by narrowing the scope of research to what standards can contribute towards enhancing flexibility and by highlighting compatibility as a frame of reference for understanding the role of standards (section 2.2.). Section 2.3 addresses the perspective taken and the sources of data used.

2.1 Refining the problem: entrenchment as lack of flexibility

In many situations entrenchment does not pose a problem. Quite the reverse. When a new LTS (-subsystem) is introduced in society, the aim is to facilitate its adoption and appropriation. Successful appropriation unfailingly leads to entrenchment – which in this respect should be understood as a sign of societal acceptance. Entrenchment is only a problem if societally undesirable features of an existing LTS cannot be transformed, a situation, however, which is difficult to foresee at an early stage of technology development.

---

\(^4\) Some official standardization sources would largely disagree with Reddy's emphasis on non-innovative standardization. They argue that the state of the art of a technology should be embodied in a standard, i.e. "the developed stage of technical capability at a given time as regards products, processes and services, based on the relevant consolidated findings of science, technology and experience." (ISO/IEC, 1991, 1.4)
development and is difficult to address at a later stage (Collingridge dilemma; e.g. Rip, 1990).

Theoretical concepts like ‘technological momentum’ and ‘path-dependency’ suggest that such changes cannot be brought about. They reflect a deterministic view on LTS evolution (Van der Vleuten, 2001), and provide few clues for policy intervention. Other concepts are more promising in this respect. In particular, the concept of de-entrenchment (Mulder & Knot, 2001), a concept set in the actor network approach, and the notion of system flexibility suggest two straightforward policy options for addressing the problem of entrenchment, namely

(1) to deal with the problem through de-entrenchment strategies that re-create the critical space necessary for change. Thereby, persuasive means to redefine the dominant problem definition of an LTS are essential to success⁵ (see Mulder & Knot, 2001); and

(2) to postpone entrenchment as long as possible and extend the system’s life span by (re-) designing flexible systems. Flexibility answers to the dual aim of public policy, namely to allow changes to existing LTSs in accordance with evolving societal interests, on the one hand, while preserving investments in LTSs, on the other. A flexible design makes systems less susceptible to premature, unwelcome entrenchment. Of interest is whether certain designs are more flexible and future-proof than others are. If so, what defines their flexibility?

The current paper addresses the second option. It centers on ways to enhance the flexibility of LTSs. Paraphrasing Feitelson & Salomon (2000, p.463), flexibility refers to the ease with which an LTS can adjust to changing circumstances and demands.

It entails openness to change.

2.2 Reverse salients, critical problems and the standards solution

Hughes system theory is not likely to bring forth the conceptual loot that adds to our understanding of system flexibility. Nevertheless, his concept of reverse salient (Hughes, 1987) well-clarifies the context in which standards enhance flexibility. He uses the term to refer to a technical or organizational anomaly that prevents the overall system from evolving. (He' draws a comparison with an advancing army during wartime: while progress is made on other fronts, one part cannot keep up with the rest and demands all attention.) In order to address the anomaly, a reverse salient needs to be translated into a problem. The difficulty is to identify and solve the critical problem that underlies the

---

⁵ Mulder & Knot’s study of PVC production identifies several strategies for de-entrenchment. These target the system’s actor network by negotiating about and redefining aspects of the critical problem (e.g. solving a different problem or assigning a new problem owner; giving in to demands with regard to one part of the LTS in order to safeguard another; and defining the problem at a higher level in order to avoid competition within the actor network at a lower level).
reverse salient (Joerges, p.13). For it is not always clear what the problem is. Moreover, usually a problem can be defined in different ways and at different system levels.

This paper starts out by assuming that lack of system flexibility is a critical problem that underlies a significant category of reverse salients in LTSs. It explores whether compatibility standards may contribute towards achieving system flexibility.

There is an intuitive tension between standardization and flexibility (Hanseth e.a., 1996) – and with reason, as the introductory paragraph shows. Consequently, the question has only been addressed incidentally and as a side-remark. For example, according to Mulgan (1990, p.202) standardization in one part of the system creates flexibility in another part.\(^6\) The idea that something interesting is at stake foremost stems from case studies. For example, the international standard for freight container dimensions (ISO/R 668) lies at the basis of intermodal transport between in particular sea, rail and road transport (Egyedi, 2000), which indicates that standards – besides working as catalysts of entrenchment - can also fulfill a flexibility-enhancing role in LTSs design. This apparent paradoxical role calls for a closer examination. The role of standards is discussed in the wider context of creating compatibility.

### 2.3 Methodological aspects

A public policy perspective is taken in this research. This is done, firstly, because important decisions regarding a significant set of LTSs – i.e. former public utilities – fall within the scope of public governance and politics, that is, within the public sphere (Summerton, 1994). Moreover, secondly, compatibility is an issue of general interest – albeit a neglected one (Egyedi, 2001).\(^7\)

As noted, there is little literature on the relationship between standards and system flexibility. To draw attention to the relationship, we organized a workshop experiment.\(^8\) Briefly summarizing the set-up: the participants\(^9\) were invited to explore whether standards can be understood as a means to increase the flexibility of infrastructures. They were asked, firstly, to draw a picture of an infrastructure they were familiar with using metaphors and analogies to stimulate free association; and, secondly, to address the following questions, i.e.

---

\(^6\) If flexibility is also to be understood as a precondition for fostering diversity, which would be a rather narrow interpretation of flexibility, David also adheres to Mulgan's point of view. "Interdependence among the development of complementary technologies may require the coordination provided by standardisation in one domain so as to foster the generation of diversity in another" (David, 1994, p.25).

\(^7\) The problem who owns the flexibility problem and in which sphere of influence solutions should be sought, can be analyzed both as an interest issue (Whose interests are served by solving the problem?) and as a governance issue (Which setting, group or actor agency controls or governs the 'inflexible' system? The market, the regulatory setting, standards bodies, etc?). Sometimes the two coincide. For example, as champion of the 'public interest' the government often formally 'co-owns' the problem of flexibility in LTSs. As partial caretaker of many such LTSs (public governance), it is also held accountable for undesirable entrenchment.

\(^8\) The experiment took place during the European Academy for Standardization (EURAS) workshop on Standards, Compatibility & Infrastructure Development, June 28th 2001, Delft University of Technology, Delft, the Netherlands.

\(^9\) The 50 workshop participants mostly consisted of standardization teachers, researchers, policy developers, and standards committee participants. 36 of them participated in the 3-hour experiment.
1. Give an example of an aspect or element that hinders the evolution of 'your' infrastructure (i.e., an example of entrenchment/lock-in).
2. How can standardization or standards aid in solving this problem (i.e., can standards re-create flexibility in the infrastructure?)
3. What new insight did you get (i.e., eye-opener)?

The drawings and answers were exhibited and presented, respectively, to the plenum in the amphitheater. Interesting insights and ideas resulted, part of which are referred to in this text (i.e. name/anonymous, Euras workshop 2001).

This paper explores in what manner compatibility standards can enhance the flexibility in LTSs. The aim is to provide insight into the behavior of standards and other compatibility strategies, and to develop a conceptual approach that clarifies in what manner compatibility contributes to the design flexible LTSs. Throughout the article examples are given, in particular from the field of ICT, a field which I am most familiar with.

The paper is structured as follows. First a closer look is taken into what the aims of flexibility can be (section 3). Next, issues of compatibility are turned to. Different types and sources of compatibility, compatibility strategies and dimensions are identified (section 4). Section 5 integrates the previous in a framework designed for finding the best match (i.e. mapping) between objectives of flexibility and compatibility strategies. Some caveats are discussed with respect to using standardization to increase system flexibility (section 6). The concluding section re-addresses the research question (section 7).

Figure 1: “Flexibility is not an end, but a means” (anonymous, Euras workshop 2001).
3. Objectives of Flexibility

“Flexibility is not an end, but a means” 10 (see Figure 1). It is a means towards what end? In LTSSs, as with other systems, several flexibility objectives may be at stake at the same time. This section explores their range. The line of reasoning is that each flexibility objective is achieved by different means - i.e. by a different kind of compatibility. Before we can make an inventory of flexibility objectives (section 3.2), we need to know from which angle lack of flexibility is regarded as a problem (section 3.1.). The angle determines the way the problem is defined and the scope of viable solutions.

3.1 Angles on flexibility: an illustration of platform-independent computing

To illustrate the significance of the angle chosen, the technical and economic angle are applied to platform-independent computing, an area that has direct relevance for the flexibility issue.

**Technical angle.** In the area of networked computing, platform-dependence is an important problem. It refers to the situation that software programs written for a certain computer platform (e.g. Ms Windows) often do not run on other platforms. They need to be re-written in order to run on other platforms as well. This would be a tedious way to overcome the problem of incompatibility - a technical ad hoc solution. In the 1990s, Sun Microsystems developed a more principled, architectural solution. 11 Its solution of cross-platform compatibility is coined as 'Write Once Run Anywhere' (WORA): a Java software developer should not need to rewrite a software program in order to have it run on different computer- or browser platforms. Together with the community of Java programmers Sun has specified a Java application programming environment which, if fully implemented by system and browser providers, meets this goal.

**Economic angle.** Viewed from the economic angle, platform-dependence signals customer dependence of specific vendors. Vendor-dependence is a sign of market failure: the market is unable to coordinate (self-govern) its players along the lines of fair competition. This failure results in decreased consumer choice and higher software prices. From the economic perspective, Sun’s response in the 1990s, i.e. the development of a Java programming environment, was not an adequate solution. Java did not answer to the problem of supplier-independence, since at that point in time Sun controlled the intellectual property rights of the Java-core, which posed its own economic dilemma. As a commercial company, Sun could –in principle - claim intellectual property rights and restrict Java licensing conditions. Whatever Sun's intentions, from the economic standpoint Sun’s technical solution represented the exchange of one type of vendor-dependence (i.e. on Ms' Windows platform) for another (i.e. on Sun’s claim on the Java-
This illustrates that contradictions arise between problem and solution if the angle taken is unclear and not consistently pursued.

In sum, although the two angles highlight aspects of the same phenomenon and are tightly interrelated, their solution to the problem of lack of flexibility differs markedly. See table 1. In the following, I pursue the technical angle interoperability standards and focus on issues that concern technical flexibility – while recognizing that in many cases system entrenchment is to an equal degree a matter of social and institutional entrenchment.

<table>
<thead>
<tr>
<th>Economic Angle</th>
<th>Technical Angle</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Problem defined as ...</strong></td>
<td><strong>platform-dependence</strong></td>
</tr>
<tr>
<td>supplier-dependence</td>
<td>software engineers: no</td>
</tr>
<tr>
<td>market players: producers</td>
<td>cross-platform compatibility, inefficient</td>
</tr>
<tr>
<td>(market monopoly / lack of fair competition)</td>
<td>engineering</td>
</tr>
<tr>
<td>&amp; consumers (lack of choice, high costs)</td>
<td></td>
</tr>
<tr>
<td><strong>Solution emphasizes ...</strong></td>
<td><strong>technology:</strong></td>
</tr>
<tr>
<td>market: conditions, regulation, etc.</td>
<td>gateways, middleware,</td>
</tr>
<tr>
<td></td>
<td>architectural change, cross-platform compatibility</td>
</tr>
</tbody>
</table>

Table 1: The relevance of the angle taken to a problem of system flexibility for the solutions sought.

### 3.2 Flexibility towards what end?

Flexibility is defined as the ease with which an LTS can adjust to changing circumstances and demands (section 2). At stake are unforeseen and unplanned requirements (Duncan, 1995, p.40) that call upon the system’s responsiveness to change. The system designer needs to have an inkling towards what end flexibility is typically needed if flexibility is to be part of the initial system design. If we look at the areas of transportation, the automobile industry and information management (e.g. Duncan, 1995; Fujimoto & Raff, 1999; Feitelson & Salomon, 2000; Byrd & Turner, 2001), some objectives seem to be common to most areas of technology. In general wordings, flexibility serves the aim of (sub)system sustainability. It is typically striven for to improve the economic rationality and efficiency of system development while preserving earlier investments in technology. Flexibility is often sought because reduces engineering efforts and eases system maintenance. In the automobile industry, for example, flexibility serves the purpose of creating a wider variety of ‘personalized’ products and services. See table 2.
Flexibility objectives:

| General: | (sub)system sustainability (change while preserving earlier investments) |
|         | i.e., technical and economic rationality |
| Specific: | reduced engineering efforts |
|          | higher efficiency of system development |
|          | reduced maintenance efforts |

Table 2: List of general flexibility objectives.

In the following, examples from the field of ICT are used to illustrate different types of flexibility objectives.\(^{12}\) Entrenchment and lack of flexibility are a familiar problem in this field. For example, since the 1970s, computer platform-dependence and lack of portability have been recognized as a problem. In large organizations, in particular, a host of entrenchment problems beset system development. “As business practices evolve, [information systems] are updated, and frequently, the resulting system grows increasingly complex, as does the maintenance process itself. (...) [The system becomes] less efficient (...) Where enhancement requires platform changes, infrastructure integrity may gradually be confounded.” (Duncan, 1995, p.43) The flexibility implied in the ideal of the 1970s, i.e. to strive for *open systems*, is still current.

The *reusability*\(^{13}\) of components and subsystems is a central feature of IT infrastructure flexibility. Duncan (1995) points to the significance of reusing information systems components for the purpose of system innovation, reengineering, and managing the rapid change of technological generations. Independent and reusable data and application components simplify the "(...) processes of development, maintenance or reengineering of direct-purpose systems", and reduce their costs (p.43).

Reusability is a general concept. Depending on the specific purpose of flexibility, reusability may refer to

- *exchangeability* (i.e. exchangeable software applications, computer hardware, etc.; reuse in a different system or context) (e.g. Dinklo, 1989),
- *portability* (refers to the different hard- and software platforms upon which software entities can operate/ be ported; reuse on different platforms), (e.g. Dinklo, 1989)
- *scalability* (i.e. the possibility to use the same software on e.g. mainframe and micro-computers; reuse in smaller/larger system) (e.g. Dinklo, 1989),
- *extendibility* or *upgradeability* (i.e. add new elements to system in order to reuse existing parts of the system and lengthen its life-span ) (e.g. Duncan, 1995)

---

\(^{12}\) Note that the aims of system flexibility may partly differ per LTS. (Egyedi & Verwater-Lukszo, forthcoming)

\(^{13}\) In the context of flexibility aims, Duncan assigns an important role to *shareability* at platform level. However, I view ‘sharability’ as an aim like flexibility and accessibility. They can be furthered by compatibility strategies but also by other means. See figure 1.
• *integration* of heterogeneous components and subsystems (i.e. reuse of part of system by integrating new elements or by integrating different subsystems; organization internally-oriented)\(^ {14} \) (e.g. Genschel, 1993), and

• *interconnectivity* (i.e. reuse of system through coupling with other (sub-)systems; organization externally-oriented) (e.g. Genschel, 1993)

• *reversibility* (i.e. reversing the changes to the system)\(^ {15} \) & *downgradeability* (likewise, e.g. for accessing an older archive)

In other words, flexibility may serve different aims, most of which can be formulated in terms of some kind of reusability. Intuitively, compatibility plays an important role in achieving these aims. This issue turned to below in more detail.

4. Compatibility

In what manner can standards contribute towards achieving the above flexibility objectives? In order to understand the workings of standards, insight is needed in compatibility. In section 4.1 and 4.2 different types and sources of compatibility are discussed, respectively. Section 4.3 discusses several compatibility strategies.

Subsystems can be compatible in two ways (David & Bunn, 1988, p.172). They can be

- *compatible complements*, that is, when subsystems A and C can be used together (e.g. plug and socket), and/or
- *compatible substitutes*, that is, when subsystems A and B can each be used with a third component C to form a productive system (e.g. IBM clones with DOS).

Compatibility is closely related to the term *gateway technology*. The latter refers to "(...) a means (a device or convention) for effectuating whatever technical connections between distinct production sub-systems are required in order for them to be utilised in conjunction, within a larger integrated (...) system." (David & Bunn, 1988, p.170) Gateways "make it technically feasible to utilise two or more components/subsystems as compatible complements or compatible substitutes in an integrated system of production." (David & Bunn, 1988, p.172) In line, the term ‘compatibility’ is in this paper exchangeable with the term ‘interoperability’.\(^ {16} \)

\(^{14}\) Reuse of part of a system for the purpose of integration with another system (part) is a transient form of flexibility: once integrated into a – higher level- system, flexibility is lost at the lower level.

\(^{15}\) I thank Harro van Lente and his colleagues for drawing my attention to the relevance of reversibility.

\(^{16}\) The term *compatibility standard* is sometimes used to distinguish this type of standard from safety and health standards (e.g. Grindley, 1995). The ISO defines *compatibility* as the "suitability of products, processes or services for use together under specific conditions to fulfil relevant requirements without causing unacceptable interactions." (ISO/IEC, 1991) It is a complex definition. Genschel (1993, pp.11-14) specifies compatibility as *interoperability* when referring to the compatibility and bridging of different technical systems. Where the focus is on elements within a system, he speaks of system integration. The line is sometimes blurred. For example, until the start of deregulation in telecommunications in the 1980's the ties between PTO's and their national equipment providers were strong. Tight vertical integration made the technical agreements between them akin to – what in other sectors would be seen as - intra-firm standards (Genschel, 1993).
4.1 Types of compatibility: Generic and dedicated gateway solutions

Gateways differ in the scope of compatibility they achieve (Egyedi, 2000). Some gateways are dedicated. They link an exclusive and specified number of subsystems. Gateways that link specific proprietary systems, such as formerly the computer networks of Digital and IBM, belong to this category. Other gateways have generic properties. Standards – and specifically those resulting from formal or consortium-based multi-party standardization – constitute a main group of generic gateways. An example is the A4 paper format. Its generic features are evident in respect to, for example, paper storage and processing devices. Even more generic are the reference models that guide standards activities where several interdependent, complementary standards are needed. One of the best known is the Open Systems Interconnection (OSI) Reference Model, which was used in the field of telematic services17. Gateway technologies can thus be categorized as dedicated, generic or meta-generic, depending on the scope of compatibility concerned.

The type of compatibility - or level of standardization - to which a gateway is submitted, determines the scope of the gateway solution. Where no standardization occurs, the connection between subsystems is, at it were, 'improvised'. This corresponds to a dedicated gateway. Standardized gateway solutions, aimed at connecting an unspecified number of subsystems, correspond to generic gateways. Gateways, which are based on modeled (standardized) solutions, that is, standardization at the level of reference frameworks, embody meta-generic properties. Table 3 summarizes the relationships.

<table>
<thead>
<tr>
<th>Type of Compatibility</th>
<th>Scope of Gateway Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>High (modeled)</td>
<td>Meta-generic</td>
</tr>
<tr>
<td>Medium (standardized)</td>
<td>Generic</td>
</tr>
<tr>
<td>Low ('improvised')</td>
<td>Dedicated</td>
</tr>
</tbody>
</table>

Table 3: Relationship between the level of standardization and the scope of the gateway solution (adapted from Egyedi, 2000).

4.2 Sources of de facto compatibility

Committee standardization18 is a means to co-ordinate the activities of parties that compete in the market (Schmidt & Werle, 1998; Weiss & Sirbu, 1990). It can be an

---

17 The OSI reference model (ISO 7498 and CCITT X.200) identifies logically separate generic functions in data communication. It depicts these as a set of hierarchically ordered layers, which address areas of standardization.

18 The term 'committee standardization' refers here to activities that are exclusively set up to lead to multi-party standards. They take place e.g. in formal standards bodies such as ISO, in professional organizations.
important step towards achieving compatibility. Ideally, the resulting standards become the shared basis for compatible implementations. However, standards do not guarantee compatibility. Whether compatibility is actually achieved, depends on the scale and manner in which standards are implemented (Egyedi & Hudson, 2001). Ultimately, public interest is not in the activity of standardization *per se*, but in compatibility and in compatible implementations. This can be achieved by other means as well. In practice, compatibility can also result as a by-product of market dominance. In the field of computer software, for example, de facto standards such as the PDF format and UNIX emerge. These sometimes more forcibly induce widespread compatibility than committee standards do.

### Table 4: Two types of specification processes that may lead to *de facto* compatibility in software. (Source: Egyedi, 2001c)

<table>
<thead>
<tr>
<th>Type of Specification Process</th>
<th>Specification Process</th>
<th>Market Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Committee Standardization</td>
<td>Multi-party Standard</td>
<td>Implemented widely? Yes &gt; <em>de facto</em> compatibility</td>
</tr>
<tr>
<td>Software Development</td>
<td>Multi-party (e.g. Open Source) Specification Market dominance? No &gt; local or no compatibility</td>
<td></td>
</tr>
<tr>
<td>In-company</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The origin of such software specifications, i.e. de facto standards, differs. See Table 4. Some result from in-company R&D efforts, others from cooperation among multiple parties. The type of specification process need have no bearing on how ownership of the specification is handled. On the one hand, a company may keep the proprietary technology for itself. It may monopolize the production of a key component, and define an interface which effectively ties complementary products of other firms to the proprietary component technology (David & Greenstein, 1990). However, on the other hand, it may also give away its technology with an eye to expected long-term advantages, or enter into coalitions with rivals to enlarge its user base and increase support for its technology. In the field of ICT, the open source approach epitomizes the cooperative alternative to ‘free market’ competition. It comes with a non-proprietary, liberal licensing regime. (The open source approach is discussed in more detail in the next section.) Whatever Intellectual Property Right (IPR-) ownership strategy used (Egyedi, 2001a), be it a proprietary or a non-proprietary one, a sizeable market share may result. If a software specification acquires market dominance, most studies retrospectively speak of 'de facto standardization' while referring to the software specification process. This confuses the issue. We should instead speak of 'de facto compatibility' to emphasize the relevance of the outcome. Compatibility is the outcome of a market-wise successful product and other multi-party fora (IEEE, IETF), or in standards consortia (e.g. W3C; i.e. multi-party industry standards fora).
development trajectory – one that may have included compatibility aims but need not
(Grindley, 1995, p.140 a.f.). This more accurately covers the source of compatibility than
presuming a direct relationship between a proprietary or multi-party standards trajectory,
the implementation of which is uncertain.

4.3 Compatibility strategies

As the foregoing already partly shows, there are several ways to address problems of
interoperability and create compatibility. The most important ones are listed in figure 4,
and are elaborated below under the headings of standardization, transparency,
compatibility artifacts, and modularity. For purpose of reference, I start with a brief
discussion of dedicated gateways, which is in most situations the default compatibility
strategy. The headings that follow refer to the dimensions that define the kind and
number of possible solutions. They show that, for example, next to ‘improvising’ and
standardizing (dimension 1: degree of standardization), there are as yet unused and
unidentified options to achieve compatibility.

Dedicated gateways

As defined earlier, this paper reserves the term ‘dedicated gateway’ for a device or
convention that, in contrast with a generic gateway - allows a limited number of
subsystems to be used together. The AC/DC rotary converter is an example of a
dedicated gateway device. In the early years of electricity, it coupled the subnetworks of
direct and alternating current (Hughes 1983). An example of a dedicated gateway
convention is the Nordunet Plug (Hanseth, 2001). This protocol provided access from
different subnetworks (i.e. OSI/X.25, EARN, DECnet, and ARPANET/IP) to a shared
backbone. Both of these dedicated gateways were designed to link specified subsystems.
Different views exist on the degree of flexibility which dedicated gateways provide.
Hanseth (2001) emphasizes the flexibility they create for experimentation at subsystem
level and their importance in the phase of system building. (The Nordunet Plug played an
important role in the building-up of Norwegian data communication.) On the other hand,
these gateways work as an ad hoc solution, often worsening the subsystem's
entrenchment, one that the system designer is trying to bridge. Although such gateways
may initially provide the required flexibility, they may turn out to be “(...) another
instance of a temporary solution to the consequences of inflexibility. (...) If gateways are
(...) [not standardized or modular] (...), they may add the sort of complexity to the
infrastructure that obstructs flexibility” (Duncan, 1995, p.49).

Standardization

Committee standards have a generic nature. Firstly, they create complements and
facilitate substitution between standardized artifacts. For example, widespread use of the
'container dimensions' standard agreed upon in ISO created flexibility in intermodal
transportation by facilitating exchange between and the use of different complementary

\[19\] In the field of ICT and in particular OSI layers, ‘gateway’ is a term reserved for application level
protocol conversion; while ‘router’ and ‘bridge’ refer to the lower network and link layers. That is, in
David & Bunn’s terminology – and my use thereof, exceptions aside – a gateway is includes routers and
bridges. (I thank Wim Vree for pointing this out.)
combinations of transport modes (*intermodal shift*). In other words, it created a technology-'neutral' system environment. The dimension of standardization depicted in the figures 2a and 3 refers to the degree of technical compatibility achieved by adopting an ad hoc, 'improvised' versus a standardized solution. (Dedicated gateways and proprietary de facto 'standards' are categorized on this dimensions as 'improvised' solutions.) Secondly, committee standards also create a supplier-'neutral' system environment, and are thus generic in the economic sense. They specify how the standardized artifact must interface and thereby create a level playing field for different system vendors. In the Information Technology (IT) sector, standards are an important weapon against supplier-dependence. Indeed, from the start of the computer era, customers have been tied to the products of their initial platform provider and could not switch systems without incurring heavy costs. Dedicated interconnections only partly alleviated the interoperability problems between proprietary systems. Although technically feasible, such interconnections were too costly, numerous and cumbersome to create and sustain. In the 1980's this resulted in standards activities which focused on 'open systems'. *Open systems* are "(...) computer environments that are based on de facto or international standards, which are publicly available and supplier independent." (Dinklo, 1989, pp.29-30)

![Figure 2a: Dimension II: Standardization.](image)

**Standardization**

"Improvised"  
Standardized

**Transparency**

Nowadays the term "openness" may refer to publicly available, shared platforms; to collaborative efforts and collective software development; and to easy access to and use of source code. That is, openness is in addition associated with availability, accessibility, collective development, 'public ownership', and transparency. Many of these elements are also features of the open source approach. Essential to the open source approach is the practice of giving free access to and sharing software source code– the disclosure of which would in former times be associated with destroying invested capital. Knowledge of the source code makes it possible to improve on the software and/or to develop compatible - complementary and competitive (substitute) - software. It eases system change and enhances system flexibility. The transparency aspect of the open source approach is a distinctive feature in respect to compatibility. In the figures 2b and 3 transparency is portrayed as a separate compatibility dimension.

![Figure 2b: Dimension III: Transparency (e.g. closed/open source code).](image)
Open source software\textsuperscript{20} generally comes with an undemanding general public license. For example, the license for the operating system Linux currently allows one to download Linux, and use, change and distribute adaptations without charge. These adaptations to the source code should, in turn, be made available in source code (see www.Linux.org). That is, with its non-proprietary, 'copyleft' approach to software development, the open source movement and the standards development community share an emphasis on public ownership. In other respects, the differences are large. Open source software may become a de facto standard, like proprietary software. In these cases, Through sheer market power, de facto standards elicit complementary compatible product development. As is the case with dedicated gateways, the technical flexibility created is limited. With de facto standards system flexibility is bound by the dominant technology.

**Compatibility artifacts**

In the following the term *compatibility artifact* is used to specify categories of technical devices and conventions that create compatibility between ICT components and (sub)systems. For example, every interface creates compatibility and is this sense a compatibility artifact. Other such artifacts are middleware, gateways and agents. The term *middleware* refers to generic building blocks that support different applications. (E.g. DirectX creates 3D images in computers games; webservices are used to communicate between applications; the Java platform is used to create a vendor-independent programming environment). *Gateways*, a term used here in the technical and more restricted sense, usually create compatibility between protocols in a fixed, static way. However, they sometimes also negotiate compatibility in a more dynamic manner. For example, Krechmer (2001) uses the term *adaptability standards* to capture the phenomenon of negotiation among standardized telecommunication services.\textsuperscript{21}

"Adaptability standards specify a negotiation process between systems which include two or more compatibility standards or variations and are used to establish communications. These standards negotiate the channel coding and/or source coding. This is an emerging form of standard. Examples include: T.30 (used with G3 facsimile), V.8, V.8bis (used with telephone modems), G.994.1 (used with DSL transceivers), and discovery protocols." (Krechmer, 2001)

The relevance of compatibility negotiation also applies to non-standardized settings. If we take a long-term view, *agent technology*\textsuperscript{22} is likely to play an important compatibility-creating role. Specific characteristics of software agents are that they are autonomous and can negotiate and interact with their environment. These features are essential to intelligent gateways. Although the technology is still largely in the research phase, in the

\textsuperscript{20} Other well-known examples are TCP/IP, SMTP, DNS and C. For more information on the open source phenomenon, see the KT&P special issue on this matter of 1999.

\textsuperscript{21} Krechmer's notion inspired the dimension of 'compatibility artifact' activity. *Adaptability standards* are defined in the following by the dimensions of 'compatibility artifact' activity as well as standardization.

\textsuperscript{22} Of the attributes of software agents, the following should at least be evident (Janssen, 2001, p.11): software agents are autonomous, goal-driven, can communicate, act towards and react to the environment.
future these agents will be designed to self-organize compatibility and manage the complexity of conversions for the sake of interoperability. In figures 2c and 3, compatibility artifacts, i.e. interfaces, middleware, gateways and agents are mapped onto the dimension of ‘compatibility artifact’ activity. This dimension identifies artifacts as being more passive or active in forging compatibility. In casu, at the high end of this dimension the artifact has the capacity to negotiate and interact in an intelligent and autonomous way (e.g. agent technology). At the low end artifacts are projects that create compatibility in a static and fixed manner (e.g. 'flat', passive interface).

![Compatibility activity diagram]

Figure 2c: Dimension I: ‘Compatibility artifact’ activity (i.e. compatibility artifacts projected on the passive-active dimension)

**Modularity**

Wolters (2002, p.263) defines modularity as follows

“A system is modular when it consists of distinct (autonomous) components, which are loosely coupled with each other, with a clear relationship between each component and its function(s) and well-defined, standardized interfaces connecting the components, which require low levels of coordination.”

In the definition the word 'standardized' loosely refers to de facto standards and in-company standards as well as committee standards. In ICT systems modularity plays a role at different system levels. For example, there are modules with software programs, but these programs may be part of what Reitwiesner & Volkert (2001) call componentware (component-based software), or, at a higher level be used as a module in pick-and-mix configurations. Modularity constitutes the fourth compatibility dimension. (See figure 2d. For purpose of overview, only the first three dimensions are depicted in figure 3.) Like the standardization dimension, on the one end of the modularity dimension the modular approach to system design is absent (i.e. 'improvised' solution is used). At the other end, modular design is part of a highly structured, architectural approach. The architecture (framework or reference model) indicates which components or modules are included and how they are interrelated.

Reitwiesner & Volkert (2001) identify two important developments with respect to modular design in the field of ICT. Firstly, the new product development strategies of
large software companies involve frameworks and componentware. The software framework coordinates the single components (i.e. pieces of software engineered for re-use), and eases the introduction of variations to products. The idea is based on the use of componentware in the automobile industry.  

Secondly, producers of standard software are making public their interfaces, enabling third parties to develop add-ons. This is an issue of transparency, which may lead to more compatible products on the market. At market level, flexibility derives from the wider possibility to pick and mix these modules into different subsets.

Figure 2d: Dimension IV: Modularity

4.4 Compatibility Dimensions

In the previous section four largely independent compatibility dimensions were introduced. Three of them are depicted in figure 3. To illustrate their overall independence, each compatibility artifact (X-axis) can be either standardized or not; designed in a modular way or not; and have its source code made accessible or not. Indeed, the modular architecture approach can also be applied to standards (e.g. OSI Reference model). Etc.

The compatibility dimensions in figure 3 draw attention, firstly, to the difference between standardization and the open source approach. From the technical angle, the open source approach is foremost of interest as a means to heighten the transparency of software and in this sense ease interoperability.

---

23 A typical application based on IBM’s San Francisco Framework consists only for 40 % of the framework and components. The rest is organisation-specific. (Reitwiesner & Volkert, p. 187) A much-used alternative is to purchase standard software. This software can be used in different domains. For this purpose one needs to " customise software to the needs of the organisation without changing source code by adjusting extensive set of predefined parameters" (Reitwiesner & Volkert, p.184). However, the flexibility thereof is much lower, the useless overhead much higher, and the % organisation specific application much lower that with frameworks.

24 The dimension ‘standardization’ is understood here not from the standard development angle – which in the case of committee standards (open, voluntary, accessible) would show overlap with the transparency dimension; rather the use of standards and standards implementations are referred to. These implementations can be more or less transparent (e.g. open or closed source XML).

25 NB1: In itself, more transparency need not increase compatibility. It can, e.g., be used to create improvised, dedicated gateways, the diversity of which would hinder overall system compatibility. As Kai Jakobs (EURAS 2002 workshop) comments, in practice, the open source approach may lead to more diversity.

NB2: Taavi Valdlo (EURAS 2002 workshop) cautions against a too theoretical view on compatibility created by transparency. In practice, the transparency of open source code is e.g. limited by the complexity of the software. If one wants to adapt open source software to one's own working environment, if a 'deeper' program level is involved it is often difficult to oversee which consequences a change will have.
Secondly, the figure draws attention to combinations and compatibility strategies that have not yet been identified. Although the majority of artifacts are dedicated 'improvised' solutions to problems of interoperability (i.e. non-standardized, non-modular means to create compatibility) with their source code undisclosed, in the future the opposite phenomenon could also become true: open and transparent implementations of software standards. The scope of compatibility strategies is wider than hitherto discussed.

![Diagram showing compatibility dimensions](image)

**Figure 3**: The three main compatibility dimensions.

Thirdly, by identifying these dimensions, it becomes easier to discuss, weigh and prioritize compatibility solutions for different situations. Each situation may need a different approach to compatibility. With respect to innovative web-developments, for example, Hanseth argues for the modular approach and the use of dedicated gateways.

“The Web technology is developing rapidly. How to integrate corporate networks and the global Internet is yet an unsettled issue. How to do this is a matter of experiment for a long time. Such experiments require modular and flexible solutions - one must be able to change the modules independently. This requires interfaces between the modules, and what is going on inside the interface is irrelevant for outsiders. Gateways are exactly the interfaces needed to make larger networks flexible.”(Hanseth, 2001)
5. Conceptual Framework for Mapping

In what manner can compatibility strategies contribute towards system flexibility? Briefly recapitulating, an inventory was made of flexibility objectives (section 3.2) and of compatibility strategies (section 4.3) and dimensions (section 4.4). These are summarized in figure 4. (See figure 4: The mapping framework.) The figure takes a bird's eye to stake out the issues of interest and prevent confusion. It indicates that, on the one hand, there may be other means to achieve technical flexibility as well, while, on the other hand, compatibility also contributes to other aims (e.g. the sharability of ICT systems).

The mapping framework aims to help (re-)design current and new compatibility policy and practices. It depicts a line of reasoning and provides concepts that are meant to help sharpen the problem definition and may serve as a checklist for reasoning through alternative solutions. Drawing on an example from the past, platform-independent computing involves two kinds of flexibility, i.e.: portability and scalability. The Java programming community addressed these problems by developing the Java platform. See figure 5. Would formal standardization of this platform, which was attempted but failed, have furthered the aims of portability and scalability? Instead of this middleware solution, can agent technology solve the problem of technical platform-dependence?

Figure 4: Framework for mapping means onto aims. Although the aim of system flexibility can be achieved by different means, the focus is here on what compatibility strategies contribute (straight arrow).

26 Nota bene: The aim was to include the most significant elements in the figure, and make an important first step towards compiling a checklist. However, as the figure indicates, it may not be complete.

27 I thank Anke Sijtsema for pointing out that traceability is also an issue in the field of ICT.
Experience and partly intuition inform us that specific flexibility objectives are best achieved by certain compatibility strategies. Sometimes there seems to be a natural link and mapping is easy. For example,

- committee standards further exchangeability;
- transparency (e.g. open source code) eases system integration;
- modularity facilitates the extension and upgrading of systems;
- interconnectivity may be well-served by dedicated gateways, but is usually better served by generic standards.

LTSs, our focus in this paper, are inherently complex. Mapping compatibility strategies will therefore be a more difficult, multi-faceted task. For example, usually the need for a flexibility system comprise more than one aim; and most likely flexibility is an issue at different system levels, requiring a combination of compatibility strategies.\(^\text{28}\) Next to clarity about the aims and means of intervention, the influence of other factors need to be acknowledged. That is, the choice of compatibility strategy also depends on other factors than the flexibility aim. For example, it is relevant to know

- whether the system environment is dynamic (if not, then a dedicated solution rather than a multi-party standard may suffice);
- for what period the solution is foreseen (if only necessary for the short term, a dedicated solution can suffice; if longer, a more durable solution such as standardization and modularity may be better);
- at what system level(s) flexibility can be achieved. Could it be achieved at different levels? Does the type of flexibility differ per level, and should the compatibility solution at these levels therefore differ likewise?
- how long the overall system is likely be useful. How important does this make the maintainability and sustainability of the system?

\(^{28}\) Note that the above discussion was still confined as far as possible to problems of technical flexibility, and does not address e.g. socio-economic flexibility, which in practice complicates matters even more.
In other words, when mapping compatibility strategies onto flexibility objectives account has to be taken of additional factors. Some main difficulties with mapping are discussed with respect to standardization in more detail below.

6. Difficulties with mapping: examples from standardization

According to participants at the Euras workshop 2001, if standardization is deliberated as a means to enhance system flexibility, certain issues are of particular interest. Firstly, it is crucial to decide at what system level flexibility is desirable, and at what level(s) standardization could contribute. This applies to all LTSs. Figure 6 uses the metaphor of the biological cell to visualize different system levels.

![Figure 6: Biological cell with different system levels. (Source: Taavi Valdlo, Euras workshop 2001)](image)

Secondly, if standardization is an option for achieving system flexibility, which part of the system, which component or subsystem should be standardized? Figure 7 uses the analogy of the bicycle to illustrate that several system components can be standardized. Often there is a choice, a choice that has consequences. As Rudi Bekkers & Heide Coenen (Euras workshop 2001) remark, "standardization of different interfaces can have a complete other set of consequences."

---

29 At the Euras workshop 2001, e.g., their relevance was explicitly mentioned for the field of ICT (Taavi Valdlo) as well as for the drinking water system (Ragna Zeiss).
Figures 7: Bicycle with different components that can be standardized. (Source: anonymous, Euras workshop 2001)

Thirdly, standards are artifacts like any other artifact. Despite their initial flexibility-enhancing effect, after a while they themselves can become catalysts of entrenchment (see introduction; Casper van der Veer, Euras workshop 2001). The more successful the standard, the bigger the problem (Ole Hanseth, Euras workshop 2001). The problem of standards entrenchment is typically felt when a new version of a widely used (de facto or multi-party) standard is introduced (Egyedi & Loeffen, 2002). In some situations standards-based islands develop. These suffer the same interoperability problem as non-standards-based islands (Egyedi, Euras workshop 2001). The entrenchment that eventually befalls all useful standards draws attention to the importance of the standard's and the system's required lifetime (see previous section). It also points to the fourth and last issue discussed here: the characteristics of the standard itself. “Standards (...) may be in place also to increase flexibility (...). Naturally, how they ensure compatibility will affect flexibility too.” (Duncan, 1995, p.54) At stake are standards' characteristics. Little systematic research has been done in this field. However, two discussions come to mind. Firstly, in the early 1980s there was discussion in Europe about product versus performance standards. The idea was that standards should indicate the requirements (aim) and, where avoidable, not how the requirement should be met. This, in order not to inhibit innovation and unfairly bias markets. For example, the ISO container standard should lay down the required container dimensions, but not which material should be used (aluminum/steel/etc.).

What other features characterize 'flexible standards'? Secondly, standards should be simple, small, and 'lightly modularized' (Timothy Schoechle, Ole Hanseth, Euras workshop 2001). According to Hanseth e.a.(1996), these features characterize the basic Internet protocol TCP/IP (Transmission Control Protocol/Internet Protocol). Hanseth et al. compare Internet with OSI (Open Systems Interconnection). They exemplify the flexibility

30 A balance should be sought in the level of standards detail (Taavi Valdlo, Euras workshop 2001).
of Internet standards as opposed to OSI standards. However, Egyedi (1999) disagrees, and argues that the OSI principle is more flexible. The core of the disagreement is summarized in Box 1. See Box 1. More thought needs to be spent on what flexible standards are, given different contexts-of-use.

Box 1: Flexibility in the OSI vs. TCP/IP debate

Hanseth et al. (1996) draw from the classic OSI (Open Systems Interconnection) - TCP/IP (Transmission Control Protocol/Internet Protocol) debate to exemplify the flexibility of Internet standards as opposed to OSI standards. However, OSI standards, in particular, were designed to be flexible. The OSI standards trajectory supported the development of a heterogeneous technical infrastructure (e.g. usable in different application and system environments). Internet standardization, on the other hand, primarily elaborated and extended its own, working infrastructure, which it did successfully. If we compare the impact of both standards trajectories on the development of the information infrastructure, in retrospect, the Internet clearly had a stronger impact. However, this says little about the flexibility of the Internet protocol. It merely indicates that in the initial stage of network development (i.e. the diffusion of network uses and the development of new services), technically flexible standards were not a pre-condition for further growth. For the full argument see Egyedi (1999).

7. Conclusion

Lack of flexibility is an acute problem for most system developers. Compatibility standards can be part of the problem, but they can also be part of the solution. Standards can enhance system flexibility. Committee standards are generic gateways. Their capacity to create flexibility is inherently larger than that of dedicated gateways and de facto standards (i.e. technical specifications that dominate the market). Standardization therefore deserves the attention of LTS designers, engineers and policy developers. In most LTSs, flexibility is sought to facilitate system development, and reduce engineering and maintenance efforts. The more specific aims differ per technology. For the field of ICT, a list of flexibility aims has been drawn up. Therein the common theme is reusability. The list includes, for example, the aims of interconnectivity, portability, upgradeability and system extendibility. They all require a certain degree of compatibility. Compatibility, in turn, can be achieved by means of dedicated gateways, multi-party standards, middleware, etc. (i.e. compatibility strategies). Four compatibility dimensions capture the set of strategies: ‘compatibility artifact’ activity, standardization, transparency, and modularity. The dimensions of standardization and modularity need little explanation: more modular and standardized artifacts usually increase the possibility to interoperate. The dimension of ‘compatibility artifact’ activity characterizes artifacts as being more passive (i.e. interface) or more active (e.g. agent technology) in forging compatibility. The transparency partly covers phenomena such as the open source
approach, which increase transparency (i.e. of software; it thereby eases interface development. Together, the first three dimensions, in particular, ease the mapping of compatibility strategies currently in use. It also raises our awareness to a potential range of future strategies, of which the most extreme example would be transparent, standardized agents.

The framework developed in this paper integrates the two sides of the argument. It lists the main flexibility objectives, and summarizes compatibility strategies. It makes explicit the need to specify the type of flexibility aimed for and weigh the relevance of each compatibility dimension. The framework provides an overview that helps system developers to reason in what respect some strategies are better matched to certain flexibility aims than others.

It is recommended that the chosen strategy be crosschecked with a number of dimension-specific – and also field-specific - issues. Where standardization is concerned, for example, clarity about the system level at which flexibility is aimed for is crucial.

Further research is needed into the dimension of transparency, into the new range of compatibility strategies implied by the four dimensions and into agent technology as an artifact of compatibility, to name a few. Many additional questions remain to be answered, e.g.: What are the flexibility objectives in other fields than ICT? What features should ‘flexible standards’ have?

References


