Which standards' characteristics increase system flexibility? Comparing ICT and Batch Processing Infrastructures

Tineke M. Egyedi [*,a], Zofia Verwater-Lukszo[b]

a Faculty of Technology, Policy and Management, ICT Department, Delft University of Technology, Delft, the Netherlands

b Faculty of Technology, Policy and Management, Department of Energy and Industry, Delft University of Technology, Delft, the Netherlands

Abstract

Most large Information and Communication Technology (ICT) systems develop in a piece-meal fashion. Their complexity and evolution is difficult to manage. They lack flexibility. This contrasts sharply with system design in the batch-wise processing industry, where flexibility has always had a high priority. In this industry, the S88 standard plays an important flexibility-enhancing role.

The paper compares the two fields of technology and explores which standards' characteristics increase system flexibility. It examines whether flexibility objectives in both fields differ, and what constitutes a 'flexible standard'.

Four standards' characteristics turn out to be important: degree of specificity, level of abstraction, system level, and degree of simplicity. They seem to be a necessary condition for standards to create flexible systems, but whether they are a sufficient condition cannot yet be said.

Keywords:

Standard, flexibility, LTSs, information technology, batch processing, S88, OSI, Internet

1. Introduction

Large infrastructure systems operate in dynamic environments. They must remain responsive to changing needs and demands, which requires active maintenance and sometimes more radical system change. However, the maintenance of such large systems, let alone their evolution, poses many problems. The Information and Communication Technology (ICT) system of a large Dutch government agency illustrates this [1]. The agency consists of several quasi-autonomous larger and smaller organisational units. Its ICT infrastructure has evolved in a piece-meal fashion. Bit by bit stand-alone, local provisions have been coupled and integrated with networked functionalities. Of the 350 software systems, 150 are generic and used throughout the organisation (e.g. word processors). Two hundred software systems serve a specific, special purpose and are only used by certain people. With respect to system maintenance, the people involved identify a number of problems. In particular,

[*]Corresponding Author. Tel: +31-15-278-6344; fax: +31-15-278-3741
E-mail address: T.M.Egyedi@tbm.tudelft.nl

The authors gratefully acknowledge Sun Microsystem's contribution towards funding this research, a contribution which made it possible to write the current paper. We also very much thank our colleagues and the participants of the EURAS 2004 workshop for their comments on our presentations.

- *the short life-cycle of IT products.* IT products have a relatively short life cycle. The average time for a software upgrade is about three years. This amount of time almost matches the time needed to roll out IT products in such a large government organisation (i.e. from idea to working implementation). As a result, there is a continuous pressure to upgrade the IT infrastructure.
- *different local needs.* Different IT configurations at the local level make it difficult to rollout IT products organisation-wide. To succeed, local adaptations are made - which further increases the differences between local configurations.
- *unsustainable software design.* Too little attention is paid to sustainable software design. E.g. software developed in a certain programming environment does not automatically run in another (user) environment.
- *unexpected interaction between software***.** New applications sometimes affect existing ones in unexpected ways[1].
- *provider dependence.* The organisation is sometimes locked into (closed source) software, such as the off-the-shelf software of a monopolist or the tailor-made software of a smaller provider. Also in the latter case system maintenance can become very dear.

The case illustrates that where information systems are updated, frequently, the resulting system grows increasingly complex, as does the maintenance process itself [2]. The complexity and further development of the ICT infrastructure become difficult to manage. The ICT system lacks the necessary *flexibility*, by which we mean the ease with which a system can adjust to changing circumstances and demands [3].[2] It entails openness to change.
Lack of flexibility is also evident in other fields of technology, in particular where Large Technical Systems (LTSs) are concerned. LTS is a term used in Thomas Hughes' system theory [4]. It tries to capture the complexity of the countless number of interrelated components and subsystems. The term comprises technical as well as *socio*-technical artefacts (e.g. institutional and regulatory provisions for artefact use and production). It includes, for example, the organizations, companies and institutions that develop around and sustain the system as well as individual actors who specialize in certain tasks, develop technical add-ons and complementary products, etc. As the LTS expands, the number of, and interdependencies between, actors and artefacts grows. Over time, these interdependencies crystallize, solidify, and make manifest a process of socio-technical *entrenchment* [5]. Changes are only possible at the cost of re-adjusting the technologies and other socio-technical arrangements that surround the LTS.[3] The higher the vested interests, the higher the costs and the more difficult it is to introduce changes to the system. A well-documented example of the problem of entrenchment is that of the environmentally harmful polyvinyl chloride (PVC) production [6]. Entrenchment fosters *path-dependency* [7]. Path-dependent system evolution cannot cope with the dynamic demands made on LTSs. Unavoidably, friction between system and its environment arises, a friction that increases as time passes.

[1] For example, unintended changes can occur when adding an application that shares the ODBC component in Windows.
[2] The term 'flexibility' is defined in management literature in terms of 'the ability of a resource to be used for more than one end product [2]". Because we take systems rather than end products as our research unit, we prefer a broader definition.
[3] We slightly adapt Collingridge's definition here. Where he emphasizes the technical, we emphasize the socio-technical character of the entrenchment process.

According to Mulgan [8], standardising one part of the system creates flexibility in another part. In this paper we pursue this idea and explore in what manner standards contribute to system flexibility. In particular, we examine *which characteristics of standards contribute to system flexibility*. We follow a two-fold approach. We analyse flexibility issues in ICTs and the role of standards therein, and compare it with the batch processing industry to gain new insights. For, possibly, different areas of technology face different entrenchment problems. Moreover, in some areas system flexibility has a high priority in system design. This is the case in the batch-wise processing industry, where the S88 standard plays a salient role in creating system flexibility. That is, we look towards the batch processing industry to explore what extra insight can be gained about using standards for designing and managing flexible systems.
The paper is structured as follows. First, we examine for what purpose flexibility is sought in ICT (Section 2). Next, we turn to the batch-processing industry and analyse the flexibility objectives (Section 3). We discuss possibly relevant characteristics of 'flexible' standards (Section 4), and explore the theoretical implications (Section 5). We conclude with some final remarks and research recommendations (Section 6).

2. Dynamic ICTs and flexibility objectives

Because flexibility is a means and not an end in itself, it is relevant to know for which system objectives flexibility is desired (i.e. *flexibility objectives*). Many areas of technology, divers as they may be, seem to share the same objectives (e.g. Duncan [2]; Fujimoto & Raff [9]; Feitelson & Salomon [3]; Byrd & Turner [10]). For example, in the field of transportation, the automobile industry and information management producers seek a kind of flexibility that allows system development while preserving earlier investments. While in the automobile industry flexibility serves the purpose of creating a wider variety of 'personalized' products, the general aim is the same as in others areas: to reduce engineering efforts and facilitate system maintenance. Table 1 lists some general, partly overlapping flexibility objectives. See Table 1.

General flexibility objectives for systems:
• to improve the system while preserving earlier investments
• to reduce engineering efforts
• to reduce operational costs
• to increase system efficiency
• to reduce maintenance efforts

Table 1: General flexibility objectives.

To examine them in more detail, we return to the field of ICT. Here, *reusability* of information system components is typically desired to manage the rapid change of technological generations. Independent and reusable data and application components simplify "(...) processes of development, maintenance or reengineering of direct-purpose systems", and reduce their costs [2]. Reusability is an overarching aim. It is an element in many of the following, more specific flexibility objectives in ICT [11]:

- *exchangeability,* that is, reuse in a different system or context (e.g. software applications, computer hardware, etc.[12])

- *portability;* this refers to the different hard- and software platforms on which software entities can run and be ported (i.e. reuse on different platforms [12])
- *scalability,* which refers to the possibility to use the same software on e.g. mainframe and micro-computers (i.e. reuse in smaller/larger system [12])
- *extendibility* or *upgradeability,* that is to add new elements to a system in order to reuse existing parts of the system and lengthen its life-span [2]
- *integration* of heterogeneous components and subsystems (i.e. reuse of part of the system by integrating new elements or by integrating different subsystems[4] [13])
- *interconnectivity*, that is, reuse of the system through coupling with other (sub-)systems (e.g. Genschel, 1993)
- *reversibility* (i.e. reversing changes to the system) and
- *downgradeability* (likewise, e.g. for accessing an older archive).

These objectives can be met by different means [1]. Standardisation is one of the most important ones, and the one we focus on in this paper. To be clear, there are different kinds of standards: standards for health, safety, etc. We restrict ourselves to compatibility (or interoperability) standards.
Compatibility between components can be of two types [14]. They can be

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

Looking at the list of flexibility objectives from this angle, the aim of exchangeability, for example, typically emphasises the substitution of software components; while 'portability' implies that software runs on different complementary platforms, which are themselves *compatible substitutes;* etc.. See Table 2 below. The Table characterises the objectives in terms of whether their flexibility exists primarily in respect to complementary or comparable, substitutive components. This answers the question in respect to which system component flexibility is sought. Perhaps even more important is the fundamental question wherein the flexibility of standards lies. The common denominator of the list of objectives seems to be the need to define component and subsystem boundaries in a way that loosens the system's fabric into reusable and adaptable units.

	Standards' characteristics	
Flexibility objectives specific to ICT	**Compatible complements and/or substitutes**	**Type of standard[5]**
exchangeability	substitutes	interface
portability	substitutes	interface
scalability	substitutes	interface

[4] 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.
[5] There are several categorisations of standards, drawn up with a specific purpose in mind (e.g. Krechmer & Baskin [20]; Sherif [21]). A selection thereof and the addition of 'procedural standards' better applies here.

extendibility/ upgradeability	complements	interface
integration	complements	reference framework
interconnectivity	complements	interface
reversibility	substitutes + complements	procedural
downgradeability	substitutes + complements	procedural

Table 2: Flexibility objectives specific to ICT characterised in terms of standardisation features.

In the last column of Table 2 we try to capture the type of standard which is typically needed to meet the flexibility objectives. In this paper three types of standards are most relevant:

- *interface standards,* which define a common interface between components;
- *reference frameworks* or architectural standards, which define how different components interrelate; and
- *procedural standards,* which define the organisational / operational procedures to (re-) create compatibility.

For example, procedural standards are typically used to be able to reverse changes to ICT systems that create unexpected difficulties.
Recapitulating, standardisation is an important means to achieve flexibility in ICT systems. The flexible property of compatibility standards lies in loosening up the interdependencies between subsystems and/or components that are of a complementary or substitutive kind.
As the example in the introduction illustrates, in the past system flexibility had little priority in ICT system design. Is this different in the batch processing industry?

3. Batch-wise processing and flexibility objectives

To respond to changes in the process industry environment, companies often have to adapt their production by manufacturing new grades or completely new products using new raw materials and new procedures. The flexible batch-wise mode of operation is recommended for such situations. Flexibility, here, can be understood as the ability of an industrial plant to reconfigure the plant equipment, including hardware and control system, and to change operational and scheduling procedures. However, this increases the complexity of management of manufacturing operations considerably. To clarify the relation between desired operational flexibility and complexity in operation management, we first describe the process industry and the way it operates.
The process industry converts raw materials into intermediary and end products - for other industries and consumers. There are two ways of processing materials: continuous and batch-wise. A *continuous process* is fed by a constant flow of feedstocks, successively runs through different process steps, and yields a constant product flow. Typically, the continuous production mode is steady state, which means that at a given point in the system there is no change in process conditions over time.[6] Examples of continuous production are the large-scale production of fuels in the petrochemical industry and of solvents in the chemical industry.

[6] Of course, during the start-up, shutdown and changeover, the production mode is different.

A *batch-wise process* is fed at the beginning by feedstocks, which then undergo a sequence of processing activities over a finite period of time. Finite quantities of material are produced using one or more pieces of equipment. Here the production mode is dynamic: to make different products the composition of the batch equipment set changes continuously. Sectors in which batch-processes typically occur are

- Food and beverages,
- Pharmaceutical products
- Paper, cardboard and paper & cardboard products
- Fine chemicals
- Rubber and plastics
- Glass, ceramics, cement, lime and plaster products
- Basic metals.

The choice between batch and continuous processing is an economic one. Investments in continuous processes are high. The process is optimised for one product, which results in a very efficient but inflexible process. For these reasons, continuous processing is used for product categories with a small product range, low product differentiation, low added value and a long production life span. Batch processing is chosen where more flexibility and less efficiency are required. Several operations may be carried out with the same equipment, and the same operation may be performed with different types of equipment. There is no one-to-one mapping between equipment and operations.
In comparison with continuous processes, batch processes are more complex and difficult to manage. This complexity stems from the non-steady state behaviour of each batch operation in itself, and from the need to align the various operations and production runs as efficiently as possible. Each batch operation runs through several phases. For example, in the case of a chemical batch reaction: process initialisation and charging of ingredients, heating (and possibly pressurizing) to establish the prescribed reaction conditions, chemical transformation, cooling of the product mix, discharging, and reactor cleaning. The frequent changing of products in a batch plant, the variability in product recipes, the sequencing problems and/or the necessity to clean the installation between batches, the dynamic character of each batch process step, etc. all make batch processes particularly difficult to manage.
It should be stressed once more, that the attractiveness of batch processing plants, despite their complexity, lies in the flexibility they offer to produce different (types of) products with the same equipment and to use the same pieces of equipment for different processing operations. These features make batch plants eminently suitable for producing a large number of product grades, short series of tailor made products and new pilot products.
The management of batch operations is so complex that a well-structured plant and process models are required. The need to standardise batch-production terminology and process modelling was first recognized by the chemical industry. As a result, in 1988, the Instrument Society of America (ISA) started a project group SP88. The first part of the standard took more than five years to complete. The ISA batch standard S88.01 "*Batch Control. Models and Terminology"* was published by the International Society for Measurement and Control in February 1995. It provided standard models and terminology for the design and operation of batch control systems. It is in this respect a reference standard.

4. Flexibility in Standards

"Standards (...) may be in place to increase flexibility (...). Naturally, *how* they ensure compatibility will affect flexibility too [2]." Duncan does not elaborate on her remark. She may be referring to technically different standards' solutions, to the way standards are implemented, but also to standards' characteristics and their influence on flexibility in technical systems. We pursue the latter interpretation, and more specifically explore whether certain standards' characteristics are more flexibility enhancing than others. Little systematic research has been done from this angle.

LTSs have a long life span. They require standards that can accommodate new demands over a longer period of time. One could therefore argue that, to avoid standards-based system entrenchment, LTSs need 'flexible' standards – or rather *robust*[7] standards, for the standards themselves do not change. As a first step towards a more systematic exploration, we focus below on standards' characteristics that can be reasoned to induce flexibility-related system behaviour. We first address a number of general, technology-independent characteristics, which are more elaborately reflected on in Section 5. We proceed with examples from the fields of ICT (Section 4.2) and batch processing (Section 4.3).

4.1 Standards' characteristics that reflect on flexibility

Four technology-independent variables (characteristics of standards) would seem relevant for standards to cope with new demands made on the system:

- *degree of specificity in standards.*
  The measure of detail in a standard can vary. Some standards only specify the required performance of a product or service (*performance standards)*, and not how to go about implementing them, as do the *product specifications*. The latter, of course, pose more restrictions on implementations. In standardisation, performance requirements are increasingly preferred to product specifications.
  To give an example, the standards for the (maritime) freight container developed in ISO TC 104 in the late 1960s were performance standards. They made no reference to the material used to put an end to early discussions on whether containers should be made of aluminium (US interest) or steel (European interest). Nor did they specify how to construct an ISO container. Their sole aim was to achieve operational exchangeability [15].

- *degree of functional inclusiveness*.
  In many cases standards include options. These options may mean different things.[8] Sometimes these options provide similar or overlapping functionalities, something that runs counter to the ideal of parsimony in standardisation. (This will be illustrated in Section 4.2.) In other cases, options represent complementary functionalities. The more functionality a standard can cater to, one might reason, the higher the degree of inclusiveness achieved. Is *functional inclusiveness* a way to cope with different and unforeseen demands? It corresponds

[7] In the context of standards *robustness* is "the ability (...) to maintain a set of specified performance features when (...) subjected to (...) external disturbances (...)." (Private communication Austine Ajah, PhD student TU Delft)

[8] Meek (1996) points out that options "come into two forms: take-it-or-leave-it, and this-way-or-that. They look like a form of overspecification, but take-it-or-leave-it can mean underspecification if one must "take it" to achieve the aims of the standard. This-way-or-that can mean underspecification if the difference between the alternatives harm the aim of the standard and the standard fails to take a position on the matter." [16]

to the ‘open systems’ vision on infrastructure flexibility mentioned by Duncan. “This vision suggests a technological capacity for ‘anything to anyone at anytime’ (...). While surely there are few (...) examples where such a capacity has been required and used, the capacity can nevertheless be defended as the epitome of flexibility [2].”
However, flexibility in the standard - system relationship depends in particular on the degree of interoperability achieved. Products that comply with different standards options may not interoperate. As will be elaborated in Section 5, parsimony of options is therefore of overriding importance for standardisation.

- *system level addressed by standardisation.*
  Large technical systems consist of subsystems, which in turn can be divided into smaller subsystems and components. Flexibility can be an issue at each level. It is crucial to know beforehand at what system level(s) flexibility is desirable and at what level(s) standards contributes most, for they may work out differently at different system levels. We therefore regard system level as a flexibility-related characteristic, too. The higher and more abstract the system level addressed, the more choices - and flexibility - in system design at the lower levels (i.e. less technically fixed).

- *level of abstraction in standards* (e.g. protocol vs. reference framework for protocol development). A standard which is based on a more abstract design approach (e.g. reference framework) incorporates more flexibility than a more detailed approach.

To test the tenability of the proposed flexibility enhancing characteristics, they are held against two empirical cases: the OSI model, a fundamental standard in ICT, and S88, a crucial standard in the batch processing industry.

4.2 OSI standard in ICT

In complex ICT systems it is not sufficient to simply define functions. It is also necessary to define the relations between these functions. Reference frameworks are developed for this purpose. The Open System Interconnection (*OSI) reference model* is such a framework. OSI was adopted as an international formal standard in 1984 (ISO 7498 and CCITT X.200). It identifies and structures all the functions needed for data communication. It depicts these as a set of seven hierarchically ordered layers. These layers represent logically separate functions of data communication. They address a more or less circumscribed area of standardisation. Standardisation of the layers can therefore take place in parallel [17].
In order to communicate between different systems, peer-to-peer protocols are defined. Each layer contains one or more protocols that specify how communication between the layer entities of both systems should proceed. One can picture the standardised data communication process as follows. In system 1 a set of data is passed downwards from the application layer to the medium. At each layer, control information is appended which indicates how the data should be processed and transmitted. The original data is, as it were, successively wrapped in different layer-specific control information. The packaged set of data enters system 2 at the lowest layer and is passed on upwards. It is successively unwrapped again until the bare set of data emerges at (or above) the application layer. To interwork successfully, the two systems must agree how to interpret the `wrappings' at a certain layer level. The software of the two systems should conform to the whole stack of layer protocols if they are to interwork.

OSI was mainly developed, first, to address lack of interoperability between systems of different vendors by defining a vendor-neutral, open technical reference framework that could serve as a starting point for system interoperability. Secondly, it was to offer a future-proof framework with which complex standardisation areas could be broken down into manageable work items [18].
Thus, *par excellence,* OSI was developed to tackle the type of LTS design problems we are examining (i.e. ex ante design of flexible, future-proof infrastructures). OSI aimed at principled internetworking solutions and provided an ex ante reference framework to guide standards activities. The resulting standards were to become the building blocks for systems that function in various networking environments. It addressed the problem of system dynamics and evolution by combining the architectural (abstract level of design) and the modular approach (independent layering, building blocks).
In terms of the four standards' characteristics mentioned earlier, OSI is a reference framework and thus incorporates a level of abstraction. It further addresses a higher system level (communication architecture between different systems) and it exemplifies performance standards in many ways. For example, it leaves the communication medium of the lowest layer unspecified (e.g. copper, air, glass fibre, etc.); it only defines which service each layer must provide to the upper layer and not how that should be done; etc. The OSI approach also shows signs of 'functional inclusiveness'. That is, the standards that were developed in the context of OSI contained many options. These options, however, mostly resulted from political compromises. A number of crucial standards showed much needless overlap between options. These options hindered end-to-end interoperability in situations where system A, which used option a, wanted to interwork with system B (using option b). Some of these options led to forks in the OSI stack [19]. That is, in this case more options would seem to hinder rather than increase system flexibility.

4.3 S88 standard for batch-wise processing

As mentioned, the S88 batch control standard tries to define terminology and concepts that make design and operation of batch-plant control easier. In principle, the standard is applicable in any production situation where flexibility is an important issue, that is, where several products can be produced with the same equipment, or the same products can be created with different equipment. Providing that finite quantities of material are produced, the models defined by the standard are applicable to batch-wise modes of operation as well as to continuous ones.
The standard describes a modular framework for recipe development. With this framework a recipe that specifies the production requirements for a specific product can be defined without support of a control engineer. Modularity allows one to reuse the process-equipment module developed for one application for another. Moreover, in principle it is not necessary to write a program for the batch control system that links recipes to plant equipment, since the product information is defined in the recipes and the equipment information in the equipment models. A recipe can be changed without re-programming the control system. Other important, partly flexibility-related advantages of using the S88 standard are:

- improved communication between all parties involved
- reduced time to reach full production levels for new products
- increased supplier-independence because of transparency
- faster and easier delivery of new products to the market
- easier optimisation of capacity usage
- reduced cost of automating batch processes

- less failures in batch control
- better maintainability of the control system
- better quality control

Not all of the models incorporated in the standard necessarily have to be implemented by a factory to adequately define control requirements. The user of the standard can decide which part of the standard is applicable to the situation and, if need be, skip part of the standard. Moreover, the standard does not intend "*to suggest that there is only one way to implement or apply batch control".* There are some degrees of freedom, as we will illustrate in the following.
The standard S88 describes three types of models, namely the process (what are we doing?), the physical (with what?), and the procedural control model (how?). They have an internal hierarchical structure. For example, in the procedural control model the procedure (the highest level) is decomposed into a set of unit procedures, which consist of a set of operations, which can further be subdivided into phases. Figure 1 presents the hierarchy of procedural elements for a fictitious example of making PVC.

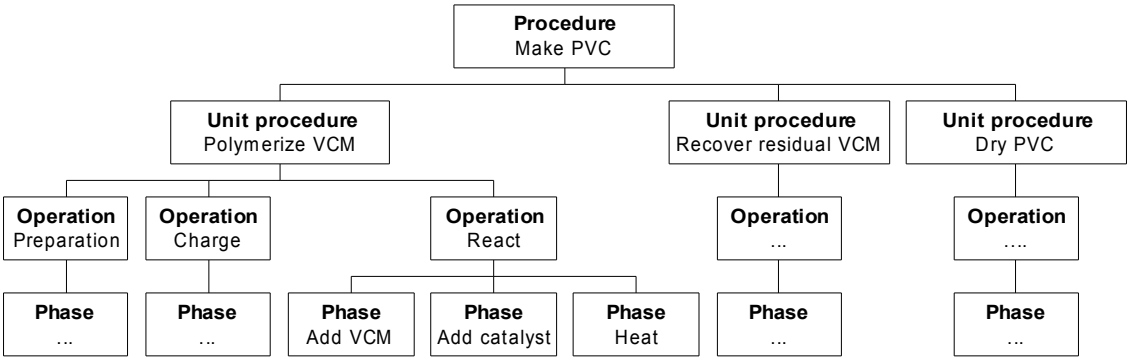

Figure 1: Procedural control model for Making PVC

As mentioned before, each recipe should be linked in a transparent way to an equipment control. The standard allows linkage at different levels of the procedural control model, i.e. at the level of procedure, unit procedure, operation or phase. Depending on the desired flexibility, the user must choose the appropriate level. The chosen level affects the system's flexibility (i.e. the ease of adapting an automated control system to changes in production recipes). To clarify, maximum flexibility is achieved by choosing the *phase* as the link level. At this level the recipes are independent of pre-defined structures in the equipment. This level is recommended for the most complex multi-product multi-purpose production situations. Its price is that the recipe should be specified to the smallest detail and therefore is time-consuming to build and maintain. In some cases, maximum flexibility is not necessary. When choosing a higher level of linkage some structures in the control systems are already predefined. For example, if the operation "charge material" is used, a predefined number of feedstocks will be charged in a pre-defined sequence. This level of linkage is preferable in situations where the operations defined by various recipes are equal and no changes are expected. The advantage of this level is that the recipes are concise and easy to create. Mostly, the pharmaceutical industry will choose for this type of linkage

because more flexibility is not required. Linkage at the procedure level is appropriate only for single-product situations (e.g. for plants producing only PVC).
The flexible standards' characteristics discussed in Section 4.1 largely apply to S88. The S88 standard is a *performance standard*: it does not describe detailed product requirements and specifies only on the desired performance. Because of this, the standard can be widely applied. It can be applied, for example, to manual batch control as well as to fully automated ones, and to batch as well as to continuous processing.
Moreover, S88 is an implementation framework (*high abstraction level)*. It does not specify the manner of implementation. The user of the standard has to fill in elements of the framework, make choices about content and decide at which system level the models of the plant should communicate with the control equipment.
The latter decision (i.e. *choice of system level*) needs some extra explanation. Choosing for the lowest level of communication means that phases of the recipe model are linked to the control equipment. It creates most flexibility in the system itself (the batch plant) as well as in the system model (the batch plant models developed according to S88). This may sound contradictory to what we remarked about the system level addressed by standardisation in Section 4.1, but actually it is not. The level to communicate with a control system refers to the implementation *phase* (how to apply the standard), while the system level addressed by the standard refers to the choice of implementation *scope* (the scope to which the standard applies). The larger the scope (i.e. number of procedures/processes addressed), the higher the system flexibility.

5. Reflection and theoretical implications

Let us recapitulate our line of reasoning. Our assumption was that, since the flexibility objectives of LTSs differ, the role of standards is also not likely to be uniform. Indeed, the previous sections illustrate that flexibility can mean different things in large ICT systems and batch processing plants. For example, there is no obvious analogy for optimising capacity usage in batch processing in the field of ICT; and vice versa, there is no evident equivalent in the batch-processing world for the ICT aim of scaleability. However, and perhaps more surprisingly, the two areas of technology also have important flexibility objectives in common such as reduced engineering and maintenance efforts, and in particular reusability. The reuse of software modules is essential to, for example, portability and scaleability. For batch processing, the reuse of equipment for different products and processes is essential for operational flexibility.
We explored which standards' characteristics are likely to be relevant in relation to system flexibility, and came up with four plausible characteristics (i.e. system level addressed by a standard, degree of specificity of a standard, the level of abstraction of a standard, and a debatable one: degree of functional inclusiveness). In two of these cases flexibility refers to a property of the standard (i.e. degree of specificity and level of abstraction). In the two other cases flexibility is related to the standard's scope (i.e. system level addressed & functional inclusiveness).
On closer inspection 'functional inclusiveness' as operationalised in terms of number of standards options is not a tenable flexibility characteristic, as the OSI case shows. The implicit assumption, that each extra option represents an additional technical functionality, is usually not the case. Standardisation is about reducing needless diversity. Moreover, the idea of 'functional inclusiveness' is in conflict with the overall argument of this article that standards may increase system flexibility at one point because they reduce options at another. If the argument applies to the overall system, it should also apply to the subsystem.

st. charact. st. trajectory	level of abstraction	simplicity (number of options)
OSI trajectory	high > more flexibility	low (many)
Internet trajectory	low	high (few) > interoperability > more flexibility

Table 3: Comparison of the characteristics of OSI and Internet standards.

In the classic OSI-TCP/IP debate of the mid 1990s, OSI standards were frequently compared with Internet standards. The latter contained few options. The Internet protocols illustrate that simplicity (few standard's options) eases achieving interoperability at subsystem level, and thus also increases flexibility at system level. This facilitated interoperability at the infrastructure level, and increased its usefulness and flexibility with regard to services that run on 'the Internet'. See Table 3.
We also need to take into account the intermediate variable of how a standard is implemented. The simpler the standard, the easier interoperability will be achieved. Interoperability, in turn, is crucial for system flexibility. See Figure 2. That is, the ease of achieving interoperability is an intermediate variable between standards and system flexibility.
In sum, although 'functional inclusiveness' may seem arguable in the light of 'flexible standards', we must discard it in the light of flexible systems. In contrast, although 'simplicity' – in terms of few standards options - can be regarded as a characteristic of an 'inflexible standard', it is a relevant characteristic for designing flexibility systems (see B in Table 4).

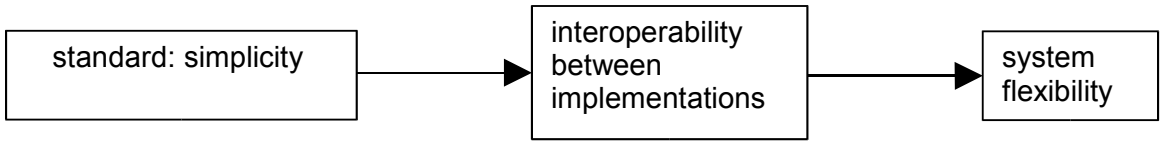

Figure 2: Relation between the simplicity of a standard and system flexibility.

Secondly, on reflection the relationship between 'flexible standards' and flexible systems is not unambiguous. As the Internet example shows, 'inflexible' standards at subsystem level can strongly contribute to flexibility at system level. Table 4 summarises the different combinations possible (i.e. from A - inflexible standards leading to inflexible systems - to D - flexible standards leading to flexible systems).
Where 'flexible standards' are concerned, flexibility is sometimes a *property of a standards category* such as performance standard (degree specificity) and standard reference framework (level of abstraction). Sometimes flexibility refers to the area covered by a standard (i.e. *standards' scope*) as is the case with the system level addressed by a standard. They represent two different angles on the relation between standards and system flexibility.

system standards characteristics	inflexible	flexible
inflexible: • degree of simplicity	A	B

flexible • degree of specificity • level of abstraction • system level	C	D

Table 4: The distinction between (in)flexible standards and (in)flexible systems

6. Conclusion

This paper addresses the question, which characteristics of standards contribute to system flexibility. The aim is to contribute towards theory that strengthens the flexible design of large technical systems. To increase the generalisability of the findings, it compares standards in the areas of technology of ICT and batch processing. Initially, four standards' characteristics are identified that seem to affect system flexibility, three of which are tenable when checked against empirical data: degree of specificity, level of abstraction, and system level. The impact of the fourth characteristic (i.e. functional inclusiveness) proves to be so strongly determined by the intermediate factor of standard's implementation that it is discarded. It is replaced by a new characteristic (i.e. degree of simplicity).
The study highlights, firstly, that where 'flexible standards' are concerned, flexibility is sometimes a property of a standards category *(property)* while sometimes it refers to the area covered by a standard *(scope)*. Secondly, a distinction should be made between (in)flexible standards and (in)flexible systems. Both flexible and inflexible standards can increase system flexibility. Thirdly, and even more important, the same standard can affect a system differently. How it affects system flexibility depends on the standards' characteristics and system level addressed during implementation.
In sum, certain characteristics of standards are highly relevant for system flexibility. This would seem to be the case irrespective of technology. However, at this stage of our research we cannot yet formulate necessary and sufficient conditions for standards to create flexible systems. Therefore, we recommend research

- that examines whether it is, indeed, important to distinguish between the two categories of standards' characteristics (property & scope) in respect to system flexibility;
- that verifies with more divers empirical material the importance and completeness of the standards' characteristics listed;
- that reviews standards' characteristics from the angle of standards implementation.

and that addresses

- the anomaly that both flexible and inflexible standards can increase system flexibility.

References

1. Egyedi TM. Trendrapport standaardisatie. oplossingsrichtingen voor problemen van IT interoperabiliteit. Delft: Ministerie van Verkeer en Waterstaat, Rijkswaterstaat/ Meetkundige Dienst, 25 September 2002.

2. Duncan NB. Capturing flexibility of information technology infrastructure: A study of resource characteristics and their measure. Journal of Management Information Systems 1995; 12(2): 37–57.
3. Feitelson E, Salomon. I. The implications of differential network flexibility for spatial structures. Transportation Research Part A 2000; 34: 459-79.
4. Hughes TP. 'The evolution of large technological systems'. In: Bijker WE, Hughes T, Pinch TJ, editors, The social construction of technological systems. New directions in the sociology and history of technology. Cambridge: MIT Press; 1987, p. 51-82.
5. Collingridge D. The social control of technology. Milton Keynes: The Open University Press; 1981.
6. Mulder K, Knot M. PVC plastic: a history of systems development and entrenchment. Technology in Society 2001: 23; 265-286.
7. Nelson RR, Winter SG. In search of a useful theory of innovation. Research Policy 1977; 6: 36-76.
8. Mulgan GJ. Communication and control: networks and the new economies of communication. New York: Guilford Press; 1990.
9. Fujimoto T, Raff D. 'Conclusion'. In: Lung Y, Chanaron JJ, Fujimoto T, Raff D, editors. Coping with variety: flexible productive systems for product variety in the auto industry, Aldershot: Ashgate; 1999, p. 393-406.
10. Byrd TA, Turner DE. An exploratory examination of the relationship between flexible IT infrastructure and competitive advantage. Information & Management 2001; 39: 41-52.
11. Egyedi TM. 'Standards enhance flexibility? Mapping compatibility strategies onto flexibility objectives,' Contribution to Responsibility under uncertainty: Science, technology and accountability, EASST 2002 Conference, Standardization Track, July 31st to August 3rd, University of York, UK; 2002.
12. Dinklo JA. Open systemen. Informatie en informatiebeleidb 1989; 7(2): 29-36.
13. Genschel P. Institutioneller Wandel in der Standardisierung van informationstechnik. Dissertation, 1993. University of Cologne, Germany.
14. David PA, Bunn JA. The economics of gateway technologies and network evolution: lessons from electricity supply history. Information Economics and Policy 1988; 3: 165-202.
15. Rowbotham M. The relationship between standardisation and regulation as they affect containerisation. In: O'Byrne L, editor. Container industry, volume 1. Proceedings of the second international container industry conference, London; 1978, p.223-234.
16. Meek B. Too soon, too late, too narrow, too wide, too shallow, too deep. StandardView 1999; 4(2): 114 - 118.
17. Tanenbaum AS. Computer networks. 4th ed. New Jersey: Prentice Hall; 2003.
18. Egyedi TM. Tension between standardization and flexibility revisited: a critique, In: K Jakobs & R Williams, editors. Standardization and innovation in information technology SIIT '99, Conference proceedings, 1st IEEE Conference on standardization and innovation in information technology, Aachen, Germany, September 15-17, 1999, Piscataway, NJ: IEEE, 65-74.
19. Egyedi TM. Examining the relevance of paradigms to base OSI standardisation. Computer Standards & Interfaces 1997; 18: 431-450.
20. Krechmer K, Baskin E. Standards, information and communications, a conceptual basis for a mathematical understanding of technical standards. In: Schoechle TD, Wagner CB, editors. Proceedings of standardization and innovation in information technology (SIIT2001). Boulder, University of Colorado; October 3-5, 2001, p.106-114.

21. Sherif MH. A framework for standardization in telecommunications and information technology, IEEE Communications Magazine April 2001; 4: 94-100.
22. Hanseth O, Monteiro E, Hatling M. 'Developing information infrastructure: The tension between standardization and flexibility'. Science, technologies and human values 1996; 21(4): 407-426.
23. ISA Standard ISA-S88.01-1995, Batch Control. Models and Terminology, ISA,1995.

Tineke M. Egyedi is senior researcher Standardization at the Delft University of Technology, the Netherlands. She has worked as a consultant and researcher in several companies and organizations. In these capacities she e.g. has done research for Dutch ministries and has participated in several European projects. Her present research projects and - coordinative tasks address issues of interoperability, standards dynamics and flexible infrastructures (Sun Microsystems, 2003-2004; EU project, 2004-2005; NGI/TU Delft, 2004-2008).
She has organized workshops and conferences on standards (e.g. IEEE SIIT2003). She is associate editor and member of the editorial board of two international journals on standardization (CSI and JITSR), and vice-president of the European Academy for Standardization.

Zofia Verwater-Lukszo is associate professor in the Energy and Industry group at the Delft University of Technology. She works on management of manufacturing operation, in particular on batch process operation. She leads the program Intelligent Infrastructures of the Next Generation Infrastructures project.
From 1998 until 2003 she was project leader of the research program Clean and Efficient Batch Process Operation, a program for developing methodologies and techniques for the batch processing industry.
She is head of the Batch Knowledge Centre (www.batchcentre.tudelft.nl) and member of the Education Committee of the World Batch Forum.