Even Desperately Needed Standards May Fail - The Case of E-Mail

Kai Jakobs
Technical University of Aachen
Computer Science Department, Informatik IV
Ahornstr. 55, D-52074 Aachen, Germany
Kai.Jakobs@i4.informatik.rwth-aachen.de

Abstract

Over the last couple of years, proactively developed IT standards were mostly failures. This includes some which were based on real-world requirements and which addressed real, urgent market needs. This is quite surprising at first glance, so this paper will try and shed some light on the reasons behind this phenomenon, using electronic mail (e-mail) as an example.

1 Introduction

Even if we disregard social, moral and religious rules for the moment, standards - in a very general sense - have been with us for quite some time. About 5,000 years ago the first alphabets emerged, enabling completely new forms of communication and information storage. Some 2,500 years later, the first national, coin-based currency, invented by the Lydians, established the basis for easier inter-regional and even international trading. The industrial revolution in the 18th century and, more so, the advent of the railroad in the 19th century resulted in a need for technical standards, which was once more reinforced when mass production generated a demand for interchangeable parts. In parallel, the invention of the electric telegraph in 1837 triggered the development of standards in the field of electrical communication technology. In 1865 the International Telegraph Union - to become the International Telecommunication Union (ITU) in 1932 - was founded by twenty states. The other major international standards setting body, the International Organization for Standardization (ISO), was established in 1947.

These days, a web of Standards Developing Organisations (SDOs) at the global, regional, and national level produce what is commonly referred to as 'de-jure' standards – although none of their standards have any regulatory power per se. Likewise, a plethora of industry fora and consortia (a recent survey found more than 260) produce so-called 'de-facto' standards almost by the week.

As a result, there exists an almost impenetrable maze of what is generally called 'standards', ranging from company specific rules, over regional and national regulations, up to globally accepted standards. Moreover, one may distinguish between different types of standards: there are voluntary, regulatory, de jure, de facto, pro-active, reactive, public, industry, and proprietary standards; this list is by no means exhaustive[1].

Indeed, it pretty much looks as if Andrew Tanenbaum was right: *"The nice thing about standards is that there are so many to choose from"*............

It is little wonder that not all standards are a success. In fact, quite a few failed, in that they were not accepted by the market. This effect may particularly be observed with IT standards. In this domain, the pace of the technical development frequently outperforms the - typically rather slow - standards setting process. After all, this process

---

[1] For a more detailed description of the various SDOs and their inter-relations, see, e.g., [1], [2].

was never designed for speed, but for the broadest possible consensus and, maybe to a lesser extent, for technical quality.

Yet, standards setting is a costly business. It has been estimated that the costs for the development of an average IT standard amount to about $ 10,000,000 [3] – and that is only one standard. Another estimation says that development cost for a 'major international telecommunications standard' may amount to some 1,000 person-years of experience, twenty person-years of actual effort, plus $3 million [4]. JTC1[2] alone has been producing between forty and fifty standards per year over the last decade.

Thus, a standard which fails to be accepted is a - possibly very painful - waste of considerable time and other resources. Accordingly, why exactly did this happen, and, ideally, how could this be avoided in the future, makes for interesting research questions. Unfortunately, this paper will not come up with answers to these questions. But maybe it is a first step in the right direction ....

The case of the X.400 electronic mail system provides us with an instructive example of a failed standard for a communication system that was desperately required by the market. This paper has a closer look at these issues and identifies (some of) the forces and developments that prevented the universally expected success of X.400.

The remainder of the paper is organised as follows: Chapter 2 will briefly provide a little background information. Chapter 3 discusses some popular beliefs regarding the failure of X.400 and the underlying OSI[3] communication protocols. The true (in my view) reasons behind X.400's failure are subsequently discussed in chapter 4. Finally, some brief conclusions are provided in chapter 5.

2 A Little Background

These days, e-mail has established itself as a convenient and virtually ubiquitous communication tool, in both private and business life. Yet, the dawn of e-mail broke only about some thirty years ago. By that time, a few simple proprietary systems were on the market, which were typically part of some larger software package (e.g., an office system). The new technology became more and popular within companies. The number of systems available on the market grew, and so did the importance of e-mail for both internal communication and information exchange with external partners[4]. A largely uncontrolled and unmanaged diffusion of e-mail systems could be observed. Eventually, many organisations, especially larger ones, had to cope with a variety of different systems, which made e-mail communication a total mess, as incompatibilities between systems led to intolerable information losses.

The widely identified need to interconnect the various proprietary e-mail systems installed at companies and other organisations triggered the initial work on what was to become X.400 in the mid to late seventies. This development started in the late seventies within IFIP[5]. Eventually, the task was transferred to the then CCITT[6] (later ITU-T), who published the first X.400 series of recommendations in their Red Book in 1984.

It should be kept in mind that at that time the telecommunication market was still regulated. The national PTTs[7] were in charge of telecommunication systems, and they met within CCITT to establish (among other things) the

[2] ISO/IEC's (International Organization for Standardization/International Electrotechnical Commission) Joint Technical Committee 1 is in charge of IT standards developed jointly by these two standards setting bodies.

[3] Open Systems Interconnection. This was an initiative in the late 70s to late 80s aiming at the development of an open architecture for communication systems. Ultimately, it failed, despite backing from all major governments. OSI was developed by ISO. Albeit originally a separate development, X.400 was widely considered part of OSI.

[4] For a more detailed account on how electronic messaging systems diffussed within companies see, e.g., [Jak ???].

[5] The 'International Federation for Information Processing'. IFIP is a non-governmental, non-profit umbrella organization for national societies working in the field of information processing. It was established in 1960 under the auspices of UNESCO.

[6] International Telegraph and Telephone Consultative Committee, the predecessor of today's International Telecommunication Union, ITU-T. This is an international organisation within the United Nations System where governments and the private sector co-ordinate global telecom networks and services.

[7] Post, Telegraph and Telephone administration.

technical specifications for such systems. There was hardly any competition at all. It was in this environment that X.400 was developed.

In the mid to late 1980s, the procurement policies of almost all major Western governments prescribed OSI-based systems. With X.400 being compatible with the OSI framework one should have expected it to thrive in this extremely favourable environment. It did not. Why?

3 Popular Beliefs

Common wisdom has it that the increasing popularity of the Internet was the one reason why OSI (and thus, X.400) never made it. Another popular explanation is OSI's 'Installed Base Hostility'. In the following I will discuss these propositions.

3.1 Internet vs OSI

"The Internet killed OSI" (including X.400) is a common point of view. And there is indeed a certain level of truth to this claim; eventually, the Internet actually did deliver the final blow to OSI. But this was only the very last step. In any case, the reasons for the growing popularity of the Internet were (are) manifold. The most obvious ones include:

- The necessary communication software[8] has been part of the increasingly popular unix operating system.
- The software has been simple.
- The software has been free.

Yet, while these facts certainly contributed to the Internet's success, it would be too simplistic to assume that they were the only ones.

At the end of the day, the Internet's standards setting process (see [5]) may be considered the main reason for its success. Its most important characteristics in this respect include particularly the evolutionary design approach, the importance assigned to backward compatibility, and a healthy degree of pragmatism.

The step-by-step approach is indeed a cornerstone of the IETF[9] process, which aims at standardising comparably small but interoperable components, which can be combined to provide the desired functionality. Most importantly, though, a certain level of pragmatism is essential. Examples of how this may work include the tendency to prefer a quick solution over lengthy discussions on merits and disadvantages of different proposals, and the rather relaxed attitude towards the use of external specifications. Both characteristics distinguish the IETF process from those of the 'official' standards bodies (like e.g., ISO), where a strong tendency to 'all-embracing', over-arching solutions that solve all problems at once may be observed. This clearly leads to sometimes extremely complex specifications, which even large companies were hesitant to implement (because of their complexity and because they tend to solve problems which nobody ever encountered). In contrast, IETF specifications are simple, and different implementations have been tested for interoperability prior to release of a standard[10].

Yet, despite these arguments, the claim that the Internet was the reason for OSI's failure (or, indeed, even a major contributor) appears to be a bit questionable (to say the least).

Just consider the individual timelines. It may seem hardly believable, but back in 1986, for instance, the Internet comprised of around 5,000 hosts, most of them located in the US. As OSI was very much a European initiative, this should have hardly been a threat at all, one should think. On the other hand, in the same year most ISO

---

[8] This software particularly included TCP/IP (Transmission Control Protocol / Internet Protocol; the technical foundadtion of the Internet), but also other protocols like SMTP (Simple Mail Transfer Protocol; the Internet's equivalent to X.400).

[9] Internet standards are developed by the Internet Engineering Task Force (IETF).

[10] This is a short and rather positive description of the IETF process. For a more critical account, see e.g., [6].

standards had either already been accepted, or were at least in a very stable state. Thus, we have to look elsewhere for other reasons for X.400's failure.

Of course, it also has to be said that the Internet has become increasingly popular since the early nineties, and has in fact marginalised OSI (which by then had already turned out to be far less popular than envisaged). This is not least due to the fact that the Internet protocol suite is readily available on virtually all major platforms, comes for free in most cases, is comparably easy to handle and does not cause major installation and maintenance problems. The base protocols are simple (newer protocols are becoming increasingly complex, though) and easy to use. Moreover, at least at a later stage, it also provided services well beyond the functionality of e-mail (as e.g. file transfer, newsgroups and, later, the World Wide Web). The exponential growth of the Internet since the mid-nineties was another major incentive to join this bandwagon. These facts were/are particularly important for the huge number of smaller companies, which have little resources and/or inadequate technical knowledge, and which do not run their own network.

3.2 Installed Base Hostility

Another issue which has been raised in the literature refers to the fact that OSI failed to provide for a smooth transition from previously used networks; it had been designed without taking into account the characteristics of older networks. That is, any transition required some form of 'jumping' [7]. In particular, X.400 is allegedly 'installed-base hostile' (see e.g. [8]). This claim is justified to some extent. X.400 was indeed 'installed-base hostile' in a way, largely due to the fact that it was considered an integral part of the OSI initiative, and accordingly initially required the use of underlying OSI protocols, implementations of which were not readily available in 1984 (and which have never been really popular anyway). This strict requirement regarding the underlying communication protocols implied that a prospective user company had to install a complete new OSI-based infrastructure if it wanted to employ X.400, a very costly exercise in terms of time and money, not to mention training and other end-user related issues.

On the other hand, the originally envisioned X.400 system, as an enabler of interoperation between proprietary e-mail systems, was certainly not 'installed-base hostile'. Quite the reverse, it was supposed to enable the single heterogeneous elements of this installed base to communicate. Likewise, when the specifications were first published there was no installed base to speak of (apart from a few proprietary systems). In particular, the Internet was little else than a network for (American) research institutions. Moreover, X.400 was designed to take advantage of the widely installed base of X.25 networks, which at that time represented the most widespread packet-switched network infrastructure (at least in Europe).

4 Why Then Did X.400 Fail?

A number of possible reasons for X.400's ultimate failure may be identified. The following sections will have a closer look at some of them.

4.1 Technical Flaws and Wrong Assumptions

There are two main reasons for X.400's problems, one technical and one non-technical. Unfortunately, they 'interworked' in a disastrous way. For one, crucial parts of the first version of the X.400 specifications were extremely sketchy, with important parts being altogether non-existent. Moreover, standardisation work ignored parallel technical developments.

4.1.1 Technical Flaws

Judging by the hectic rush that surrounded the publication of the first X.400 specifications in 1984 it would appear that whatever was available at the time had to be published, regardless of timing and quality. In a way, this is understandable. At that time, CCITT's work was organised in four-year intervals, called 'study periods'. Standards (referred to as 'recommendations') were published only every 4 years, at the end of a study period.

Thus, if another four year delay was to be avoided, something had to be published by the end of the 1980-84 study period.

Yet, with hindsight this may not have been the best solution. The technical shortcomings of the first specification resulted in a massive reduction of both the system's user-friendliness and its capability to be integrated into existing IT environments. For one, the first X.400 specifications were technically premature. In fact, crucial parts of this first version were extremely sketchy, such as e.g., the security features and directory interworking). Other features were altogether non-existent (as e.g., the message store and interworking with the directory service). They were added at a later date, i.e., the 1988 version was much more complete, but apparently it was too late then.

Moreover, X.400 was first published before other OSI standards, which were crucial for its proper functioning, were ready (this was corrected in the second version in 1988, but not without considerable difficulties). It may well be assumed that implementations based on these inadequate initial specifications contributed heavily to the less than satisfactory utilisation of X.400 based services (see sect. 4.1.3 below).

4.1.2 Wrong Assumptions

In addition to the above technical issues the X.400 specifications suffered from the enormous speed of technical development in the IT domain, and from the fact that such technical developments which occurred in parallel with the X.400 standardisation process were ignored.

Technical work on the specifications started in the mid-seventies. At that time, 'dumb' terminals, typically connected to a mini computer or a mainframe, were the prevailing end-systems available to the end user. Consequently, an environment was assumed that was built around this type of technology that represented state-of-the-art in the late seventies to early eighties. A technical detail highlights this problem: the initial specifications did not include a 'Message Store' (MS). An MS would allow to store (and retrieve) messages permanently on a machine, even if it is switched off. Assuming that larger machines (minis/mainframes) were used implied the assumption of continuously running machines (these types of machines were hardly, if ever, switched off under normal circumstances). Yet, the diffusion of PCs meant that more 'intelligent' end-systems became available which would, however, normally be switched off at the end of a working day. They would have required a message store. Thanks to its design, adopting X.400 to this new environment was less than trivial, and wasn't really attempted at all for quite a while.

Maybe even worse, X.400 suffered from a paradigm shift during ist design. Initially X.400 had aimed at interconnecting the huge number of different proprietary e-mail systems that had been implemented at many sites. The idea had been to establish a standards-based 'backbone' network that would be able to interconnect all different types of other e-mail systems, regardless of their respective technology. However, during the course of the work this changed. The new paradigm was that X.400 would be the ubiquitous e-mail system, providing its functionality to the end-user's desktop.

In fact, this shift is likely to be an important contributor to X.400's problems in the market. Here again technical progress overtook standards development. By the mid-eighties, LAN[11]-based e-mail systems had become the systems of choice for internal communication in virtually all organisations. Not unlike PCs, such LAN-based systems did not really fit into the assumed X.400 environment.

Taken together, these two developments - the diffusion of PCs and LANs in the mid to late eighties - rendered the - new - idea of 'X.400 to the desktop' virtually obsolete. In more general terms, the time span between the start of the standards setting activity (preliminary work started in the mid/late seventies) and the completion of the final documents led to a missed window of opportunity. Other systems (i.e. PCs and LANs) had occupied the major market segment of corporate internal communication systems. Somewhat ironically, this left X.400 with the backbone market for which it had been intended in the first place.

---

[11] 'Local Area Network', a comparably fast network, typically installed on its operator's premises (e.g., within a single building or a campus).

4.1.3 The Effect of Poor First Implementations

Especially in the fast-moving IT domain uncertainty affects all predictions, and it has a particular strong impact on standardisation. Here, big oaks from minor acorns may grow. That is, comparably small events may carry great weight; in the absence of a sound basis for judgement and decisions the adoption of a particular technology by just one firm may encourage others to follow suit. If this happens, chances are that an inferior technology will be adopted, which may suit the initial adopter (who will have evaluated the alternatives and selected the technology to best suit his needs), but does not necessarily meet other entities' demands. They, in turn, will then make their choices solely based on the initial adopter's policy decisions. Little, if any experimentation with alternative technologies or systems will occur, which will rapidly be discarded. A similar effect may be observed when decisions to adopt are based only on initial expressions of a technology (e.g. implementations of IT systems). In such cases, a poor first implementation can easily reduce to zero this technology's chances of being adopted, since possibly superficial, implementation-specific shortcomings hide the technology's inherent advantages (for example, X.400 suffered from many inadequate early implementations). Likewise, observable early benefits of a technology will outweigh all other aspects; in particular, higher benefits to be gained from a different technology at some later stage will be ignored. Indeed, these benefits again cannot be identified at all due to the lack of opportunities for experimentation. It follows that the market can - and frequently will - adopt the 'wrong' technology when left on its own. 'Wrong', like 'right', is of course a vague term; a technology may appear to be 'right' for a particular adopter, but at the same time the adoption may have negative impacts on others [9].

4.1.4 National Monopolies and the Standardisation Process

Finally, another aspect should be mentioned here. Although the initial specifications failed to provide for several important features X.400 systems have always been considered as extremely complex and hard to manage. Indeed, X.400 aimed at providing the one solution to all e-mail related problems. As all voting members within CCITT committees came from PTTs[12] or equivalent organisations - who at that time still enjoyed a monopoly situation - it comes as no big surprise that they did not follow a more user-friendly, gradual approach, with a first specification evolving along with upcoming new requirements. Rather, they were in position to follow a take it or leave it approach, and design a system that clearly reflects PTTs' ways of thought and that met their specific needs, as opposed to those of their users.

Despite the undeniable general need for open, vendor and platform independent communication, the developers of X.400 apparently failed to realise that a system as complex as this, operating on top of an equally complex protocol stack will be useful only for a handful of large, technically sophisticated organisations. Accordingly, OSI must be considered a failure in the market place even although it correctly anticipated general initial requirements.

5 Some Conclusions

There was a clear need for a standards-based electronic mail system in the late seventies - mid-eighties (and beyond). X.400 was an attempt to standardise such a system, initiated by the CCITT, with backing from virtually all Western governments. Still, it never really got off the ground.

It has turned out that the major problem behind this failure was not the success of the Internet (which was little more than a US research network by that time). Rather, the standards committee's negligence with respect to technical developments which took place in parallel with X.400 the specification work needs to be blamed. As a result, X.400 could not be integrated into the technical environment that was state-of-the-art when the first specifications got published. To make things worse, the original idea underlying X.400, i.e., to create a backbone e-mail system to interconnect the numerous proprietary systems that had hit the market was abandoned in favour of the idea of a ubiquitous e-mail system. This made integration of X.400 into existing IT environments

[12] Post, Telegraph and Telephone administrations; the (then) monopoly organisations were in charge of the respective national (telephone) networks.

extremely complex. In contrast, integration of the originally planned backbone would have caused comparably little problems (and may have led to a successful system).

As a consequence, we may state that standards setting in the IT arena, were technical developments continue to happen at a very high speed, need an adequate level of flexibility to be able to react to outside developments that are likely to affect the system to be standardised. And the second (albeit old for some) lesson would have to be that all-embracing, over-arching, and therefore very complex standards carry a greater likelihood of failure than modular, extendable and thus more flexible ones.

6 References

[1] Cargill, C.: "Consortia and the Evolution of Information Technology Standardization", K. Jakobs, R. Williams (eds): *Proceedings SIIT '99*, IEEE Press, 1999.

[2] Jakobs, K.: *User Participation in Standardisation Processes - Impact, Problems and Benefits*. Vieweg Publishers, ISBN 3-528-05689-4, 2000.

[3] M.B. Spring et al., "Improving the standardization process: Working with bulldogs and turtles.," B. Kahin, J. Abbate. (eds): *Standards Policy for Information Infrastructure*, MIT Press, 1995, pp. 220-52.

[4] Office of Technology Assessment (eds), *Global Standards – Building Blocks for the Future*. OTA-TCT-512, 1991.

[5] Bradner, S. "The Internet Standards Process -- Revision 3". *Internet RFC 2026*, 1996

[6] Jakobs, K.: "Trying to Keep the Internet’s Standards Setting Process in Perspective". *Pioneering Tomorrow’s Internet*; Proc. TERENA Networking Conference, 2000.

[7] Hanseth, O.; Monteiro, E.: "Changing irreversible networks. Institutionalation and infrastructure". *Proc. ECIS 1998*, Provence, June 1998.

[8] Hanseth, O. et al.: "Developing Information Infrastructure: The Tension Between Standardization and Flexibility". *Science, Technology & Human Values*, vol. 21, no. 4, 1996.

[9] Besen, F.M.: "The standards process in telecommunication and information technology". Hawkins, R.W. et al. (eds), *Standards, Innovation and Competitiveness*. Edward Elgar Publishers, 1995.

Short Bio Kai Jakobs

Kai joined the Technical University of Aachen's (RWTH) Computer Science Department as a member of technical staff in 1985. Since 1987, he has been Head of Technical Staff, Chair of Informatik IV (Communication Systems). He holds a PhD in Computer Science from the University of Edinburgh.

Earlier research interests and activities concentrated broadly on the user-friendliness of data networks. This included, for instance, support of group communication, user-friendly naming, directory services, and services to be provided by the underlying infrastructure. More recently, interest shifted to various aspects of standards and the standardisation process, focussing on the individual's role in standards setting, and on the pros and cons of user participation in this process.

Kai is (co)-author/editor of a text book on data communication and, more recently, three books on standardisation processes in IT. More than 100 of his papers have been published in conference proceedings, books, and journals. He was on the programme committees of numerous conferences, and has also served as an external expert on evaluation panels of various European R&D programmes, on both technical and socio-economic issues.