

## Why Java™ Was - Not - Standardized Twice

Tineke M. Egyedi

*Information and Communication Technology  
Faculty of Technology, Policy and Management  
Delft University of Technology  
T.M.Egyedi@tbm.tudelft.nl*

### Abstract

*Proprietary de facto standards are seldom formalized. This paper examines a case, the Java™ Technology of Sun Microsystems, where this was attempted. Sun approached the ISO/IEC JTC1 standards body and later the ECMA standards consortium to formalize Java. It withdrew both times. In this paper, I examine what motivated Sun's actions. A conceptual framework is applied that distinguishes two levels of coordination in standardization: 'technology-oriented compatibility control' and 'orchestration of market orientation'. Sun's actions addressed both levels. It initially used standardization to focus attention on Java™ and increase confidence in an open, stable Java specification process. But it turned to proprietary 'compatibility control' in reaction to standards politics and developments in the market.*

### 1. Introduction

When Sun Microsystems approached the ISO/IEC Joint Technical Committee 1 (JTC1) to standardize its Java™ Technology in 1997, Java was already well on its way to become a *de facto* standard. Sun became a recognized Publicly Available Specifications (PAS) submitter late 1997 but refrained from using its submitter status, allegedly because JTC1 had changed the PAS procedure. In April 1999, Sun approached the ECMA standards consortium, an international industry association for standardizing information and communication systems, for the same purpose. If Java became an ECMA standard, it could be submitted to JTC1 by way of the Fast Track process. However, after the first meeting of the ECMA standards committee Sun again withdrew. This time ECMA's Intellectual Property Right (IPR) rules were not elaborate enough, according to Sun. Two main questions arise. Firstly, why did Sun initiate formal and consortium standards activities in the first place? Secondly, why did Sun pull back twice?

There is a host of literature that addresses why companies partake in standardization. Standardization is part of the competitive product development process between producers [1,2]. Companies partake in order to develop new markets and protect established markets (e.g. prevent compatibility to block competitors from their market). They use standards as change agents. They use them as strategic tools to consolidate a market position or gain advantage over competitors [3,4]. This body of literature suggests that dominant market players, whose products have become a *de facto* standard, have few incentives to standardize. They are more likely to withhold information on interface specifications or change proprietary product interfaces at regular times to put off competitive product development. Or they may try to tie complementary products of other firms to their proprietary component technology. With an eye to long-term advantages, they may give away a technology or enter into coalitions with rivals to enlarge their user base and widen support for their proprietary standard [5]. However, the step towards formal standardization is seldom taken. In this respect, the initiative to standardize Java™ seems to be rather unique<sup>1</sup>.

This paper is a case study. Its aim is to explore why a company would want to formally standardize its *de facto* standard and under what circumstances it may withdraw again. In section 2, I develop a conceptual framework that helps me interpret the Java case. Next, I provide the necessary background to the Java technology and discuss Sun's main compatibility strategies regarding Java (section 3). Sun saw two ways of getting JTC1 to recognize Java: the straightforward route via the PAS procedure and the Fast Track process through ECMA. I describe them in

---

I gratefully acknowledge the comments and suggestions of Jaap Hoogenboezem and four anonymous reviewers on the earlier draft. This research received financial support from the European Commission, DITSE/ Delft University of Technology, and Verdonck Holding BV. They need not agree with and are in no way responsible for the views expressed in this paper.

<sup>1</sup> A second exception is Adobe's PDF-format, which was offered to ISO [6].

section 4. In section 5 and 6, I address what motivated Sun to initiate and later drop its standards activities in JTC1 and ECMA, respectively. I compare Sun's strategies in both situations in section 7 and draw some conclusions.

## 2. Conceptual framework

The literature mentioned in the introduction discusses some of the motives of companies to participate in standardization. They show that standardization is an endogenous factor in market development. Company motives and strategies regarding standardization are generally not openly discussed. Most likely, a company will adopt a set of complementary strategies and adapt them according to the circumstances. Its strategies and the circumstances determine whether a company standardizes or not, whether it chooses a consortium or a formal standards body to do so, and - if a choice exists - which technical committee is most likely to incorporate its proprietary solution in a standard. The legitimate, formal rationale to approach a standards body is to seek technical compatibility among products and services. The resulting committee standards aim to coordinate the activities of product and service developers [7]. But company strategies may also aim at another level of coordination. In economic studies, strategies are noted such as compatible product pre-announcement [8] and 'embrace-and-extend' [9]. These aim to direct the actions and orientation of other market players by means of standardization. They address the strategic level of market coordination and complement the level technically oriented compatibility strategies. Both aspects to committee standardization serve as an *ex ante* market mechanism.

Company strategies may support both levels of coordination. Usually, however, some strategies emphasize technical compatibility while others focus on orchestrating market orientations. I conceptually distinguish the first category as referring to *technology-oriented compatibility control* strategies. As will be discussed in section 3.2, Sun has actively used instruments such as licensing, test suites and the Java Community Process to safeguard the development of compatible Java implementations. In many respects, Sun's concern for compatibility resembles that of standards bodies. Compatibility requires the coordinated development of a set of specifications and consistent implementations. These technology-oriented aims were essential to Sun's market.

The second category of strategies consists of market politics inspired *orchestration of market orientation* strategies. At a conference that preceded the first ECMA committee meeting, Sun's director of standards, Carl Cargill, remarked that companies which dominate the market have no inclination to standardize because

standardization would mean opening up the market for other players. Since Java was by then a *de facto* standard, did Cargill's remark imply that Sun embarked on ECMA standardization because it felt threatened by other market players? The most obvious way to counteract competing developments and prevent fragmentation of the Java market is to involve competitors in developing Java specifications. This could be organized within Sun's own forum, the Java Community Process. However, such a proposition is likely to be met with distrust. In line with its announcement at a very early stage of Java development (e.g. JavaOne conference 1996), that it intended to standardize Java, the step towards recognized, consensus-driven formal or consortium standardization may have appeared a more effective strategy. The promise of standardization would in this scenario be a means to re-focus the orientation of competitors towards a Sun-driven initiative.

The two levels of coordination that underlie the types of strategies discussed above are used as complementary explanatory frameworks. The distinction allows me to highlight changes in the strategies used by Sun - although specific strategies may sometimes address both levels of coordination. The first explanation emphasizes Sun's interest in compatibility, which coincides with the interests of Java programmers in standardizing Java - an accepted, legitimate standardization aim. The second one centers on Sun's strategies to influence the position of market players and focus their activities. These alternative explanations are used to analyze Sun's initiation and withdrawal from JTC1 and ECMA standardization. The research design is summarized in Table 1.

**Table 1. Research design**

Standardization actions> Coordination strategies	Initiation	Withdrawal
Technology-oriented Compatibility Control	?	?
Orchestration of Market Orientation	?	?

The data used for this research stems from formal interviews and informal conversations with experts<sup>2</sup> and participants to the ECMA Technical Committee 41 (TC41) meetings, my observations during these meetings and analysis of the accompanying TC41 email exchange, and a study of the relevant documents, press releases,

<sup>2</sup> Jan van den Beld (Secretary General of ECMA), Willem Wakker (ACE Consulting), and Roger Martin (Sun standardization strategy manager), whom I very much thank for the interviews. They may not agree with my interpretations of the events.

web-based articles and comments on Sun's activities in JTC1 and ECMA (e.g. discussion on mailing lists).

### 3. Sun-driven Java developments

Some background information on Java and Sun is necessary to understand Sun's initiative to standardize and the reactions it evoked. I therefore briefly introduce the technology and discuss Sun's most influential compatibility strategies. (For a more elaborate discussion, see [10])

#### 3.1. Java Technology™

Java originated from a Sun research project in the early 1990s. The idea was to connect different computer-based devices (e.g. household appliances, television sets, etc.) in a network. Sun developed a computer language to work in these devices. The language was simple and concise (kilobytes, not megabytes) and network-oriented (no software in the devices). Its design addressed heterogeneous architectures, software portability, and safety. The project was abandoned because there appeared to be no market. But in 1995 Sun realized that the project outcomes could be used for programming for the Internet. They called the language Java. Its platform-independence allowed small Java programs to be downloaded and executed by web browsers. These moving, colorful applets triggered Java's breakthrough on the Internet.

One of Sun's maxims was 'Write Once Run Anywhere' (WORA): a Java software developer should not need to rewrite his or her software program for different platforms. Java programs were to be portable and scaleable. In order to achieve cross-platform compatibility, Sun created a standardized application programming environment. Each system and browser provider was to fully implement the specifications and Application Programming Interfaces (APIs)<sup>3</sup> of the standardized Java environment if WORA was to be achieved.

With regard to the Java programming environment, the Java Programming Language remained stable. Over the last few years the number of APIs strongly increased.<sup>4</sup> Several system providers, such as IBM and HP, developed compatible Java Virtual Machines (JVMs, i.e. software that runs on proprietary operating systems and is capable of interpreting compiled Java byte code). The Java™ technology that was considered for standardization consisted both times of the Java Language Specification,

the Java Virtual Machine Specification, and the Java API Core Class Library Specification.

Java is presently also used for purposes that Sun originally had in mind. This area of application is called embedded Java, or real-time embedded Java. The focus in this paper is on the Java programming environment and only addresses real-time developments in so far as the latter affect the former.

#### 3.2. Compatibility strategies

Cross-platform compatibility was essential to WORA and therefore to Java development. Sun used several means to ensure compatibility. The most significant ones are listed in Table 2. Sun started by giving interested parties access to its source code. It invited developers to comment on, experiment with and improve the original source code. Throughout the years, the Java source code remained available.<sup>5</sup> But, from the start, Sun retained control over the process. The source code was 'open' in the sense of being accessible and free of charge, but, for example, the decision about changes to the original code lay in Sun's hands and commercial use was bound to license restrictions.

One means to foster and maintain compatibility was Sun's licensing policy. Part and parcel of this policy were the test suites which were used to certify compatible Java products, and the Java-compatible logo (the steaming cup of coffee) to brand compatible products. These instruments of control were closely tied to Sun's ownership of and IPR to trademarks (e.g. Java™ and Java Compatible logo), patents (software algorithms) and copyright on the specifications. Pressed by its commercial licensees, Sun developed a 'Community Source' licensing model, which sought to combine the advantages of the Open Source licensing model and the Proprietary licensing model [12]. It did, indeed, represent a more liberal licensing regime for commercial parties, but Sun still retained ownership of the original code, the upgrades, and the test suites.

**Table 2: Compatibility strategies used by Sun**  
(source: adapted from Table 2 in [10])

Coercive strategies	Forceful strategies
---------------------	---------------------

<sup>3</sup> APIs comprise the standard packages, classes, methods and fields made available to software developers to write programs [11].

<sup>4</sup> Some say the time to market has been set too short because the current set of APIs contains many bugs. [26]

<sup>5</sup> For example, early 2000 the source code of the Java™ 2 platform Standards Edition was made available.

<ul style="list-style-type: none"> <li>• (quasi-) Open Source Code</li> <li>• Instructional books on Java</li> <li>• Certified Java training programs</li> <li>• Distribution of the Java Software Development Kit</li> <li>• ANSI/JTC1 standardization</li> <li>• JTC1 PAS procedure</li> <li>• ECMA/JTC1 Fast Track procedure</li> </ul>	<ul style="list-style-type: none"> <li>• Java Community Process</li> <li>• Java Specification Participation Agreement</li> <li>• Licensing</li> <li>• Technology License and Distribution Agreement</li> <li>• Sun Community Source Licensing model</li> <li>• Reference Implementations</li> <li>• Test suites</li> <li>• Java Compatibility logo</li> </ul>
--	---

A Java community existed, which Sun formalized in December 1998. In the Java Community Process (JCP) manual, Sun unfolded "(...) a formal process for developing Java™ specifications (...) using an inclusive, consensus building process that not only delivers the specification, but also the reference implementation and its associated suite of compatibility tests." [13] The procedures were criticized for representing a 'gated' community process. Sun's role was too dominant and independent Java developers had no influence [14,15]. In the Spring of 2000, Sun distributed a second version for public comment [17]. It differed in many ways from the first version and answered to much of the critique [16] The second version assigned an important role, firstly, to the Executive Committee, which represented "major stakeholders and is responsible for approving the passage of specifications through key points of the JCP (...)" It consisted of 16 members and a chair. Two of them were Sun employees. (See appendix A of [17].) Secondly, the 'Specification Lead', that is, "the person responsible for leading the effort to develop or make major revisions to the specification", would be influential. But where changes to the Java-core were proposed, the Sun representative still needed to cast an approval vote.

The idea of WORA and Sun's strategies to involve others in developing and implementing the Java platform led to a large user base. In 1999, there were more than 1,3 million Java developers (International Data Corporation, op. cit. in [18]). This figure consists of developers who work for companies and a majority of independent developers.

#### 4. JTC1's PAS and Fast Track procedures

ISO/IEC JTC1 has several procedures to ease the processing of externally developed standards. Examples are the Fast Track process (1987) and the PAS procedure (1994/1999). Both procedures are relevant to Sun's standardization initiatives.

The Fast Track process is an option for consortia and other multi-party fora that have an A-liaison membership

status in JTC1. The A-liaison status is meant for organizations that contribute actively to JTC1 standards committees (e.g. ECMA and IEEE). It gives access to the Fast Track procedure: an A-liaison member can submit its specification as a final Draft International Standard - and thus skip the prior phases of the JTC1 standards process. This procedure strongly reduces the time needed for standardization. ("The duration of the final ballot, to become an IS ballot is six months." [20])

The procedure for the *Transposition of Publicly Available Specifications into International Standards* is based on the Fast Track process [19]. It also allows an external organization to submit its specification as a draft International Standard. (JTC1 aims to complete the transposition in 11 months [19].) But the criteria for becoming a recognized PAS submitter are less restrictive than those for an A-liaison membership. "It is expected that these procedures will be used to process a broader class of documents from a more diverse set of sources than is currently served by the Fast Track process." [20] For example, JTC1 originally installed the procedure to formally standardize Internet standards among which TCP/IP, the most important one [21]. In 1994 the Internet Society (ISOC) first sought an A-liaison with JTC 1, expecting that peer level recognition would encourage Internet's acceptance by governments. However, because TCP/IP competed with Open Systems Interconnection standards (OSI), ISOC could initially only apply for an A-liaison by pursuing the convergence path. Convergence was not required to become a PAS submitter.<sup>6</sup>

#### 5. JTC1, the first attempt

Sun was the first private company to apply as a recognized PAS submitter. IBM strongly backed up Sun's application. This happened in March 1997. It caused a stir, because although the rules allowed individual companies to apply, the criteria favored open, consensus-oriented organizations. In July, Sun's application was turned down with comments. The comments of the JTC1 national members roughly focused on Sun's desire to keep the Java trademark for itself and have the JTC1 standard called something else; on what body would be responsible for updating and maintaining the Java standard; and on whether Sun would be open in accepting changes to the standard [22]. Sun addressed the comments in September 1997 and reapplied as PAS submitter [23]. It suggested, for example, that a JTC1 working group, which would be open to all stakeholders, would address the standards maintenance work, and it offered to supply the project editor. Two months later, Sun was accepted as a PAS

<sup>6</sup> Ultimately ISOC did not become a PAS submitter. Instead, a co-operation agreement was drawn up with JTC1 SC6.

submitter. But, again, there were comments [24]. The national bodies expected their comments to be addressed in the Explanatory Report that would accompany Sun's submission of the Java specs, and they added that voting 'yes' at this stage did not automatically include approval of the specs.

According to Sun, the positive outcome of the voting was to be understood as international approval of Sun's open Java development process. In the following year, Sun did not take steps to actually submit the Explanatory Report or the Java specifications to JTC1. Sun silently withdrew from the PAS process, a move that became apparent when Sun's overtures to ECMA became public.

### 5.1. Initiative

In the following sections, I try to distinguish between Sun's explanation of the events and my interpretation of them, because they do not always coincide. I use the headings of 'stated reasons' and 'interpretation' for this purpose.

*Stated reasons.* Sun said its goal always was to "have Java, already a de facto international standard, codified as a *de jure* standard" [25]. From a business perspective, Sun's interest in standardization was to increase the visibility and importance of Java and to promulgate a network-centric view on ICT developments. By approaching JTC1, Sun signaled that Java was to be a specification that people could rely on as being stable and that it would not be changed unexpectedly. It allowed people to make a commitment to it.

Sun chose the PAS procedure because this was the most effective way to get the Java technology formally accepted worldwide. It was a means to get easier access to the public procurement market, and to preserve industry's substantial investment in Java. The latter argument can be understood as a way of saying that the Java submission should not undergo serious changes during the PAS review process.

*Interpretation.* Sun did not intend to hand over the evolution of Java to JTC1 [27]. It expected to retain control over the standards maintenance process by safeguarding the role of the Java community during JTC1 standardization, whose input that was coordinated by Sun itself. ("The JTC1 working group that will address standards maintenance must be responsive to international Java community." [23]) Sun upheld essential IPRs, and retained its patents (although no fees are asked), its copyright (joint-copyright ownership was suggested, no fees asked), and trademarks (e.g. control over compatibility logo). An additional benefit of the PAS procedure was that ongoing Java developments would become tightly linked to standards development. The revenues from IPRs were forfeited in exchange for

enlarging and stabilizing the Java market - without compromising control over cross-platform compatibility (e.g. by means of the Java compatible logo and the test suites). JTC1's role was to codify and ratify the specification development activities supervised by Sun.

Sun's PAS initiative can therefore best be understood as a means to 'orchestrate the orientation of market players'. There are two main reasons to think so. Firstly, because JTC1 was the pre-eminent international standards body for IT matters, it was a focal point for consensus-based standards development. The PAS procedure would appear to leave room for the influence of competitive market players, keep them oriented towards Java developments led by Sun, and dissuade competitive developments.

Secondly, in the years that preceded the PAS initiative Java was becoming a hype (1995-1996). Mainly by way of Netscape Navigator, copies of Sun's Java runtime environment were downloaded to the PC systems of Windows users. Sun's network-centric vision and Java's promise of platform-independence made Microsoft nervous. Sun was challenging the basis of Microsoft's software market, the Windows platform. In 1995, Microsoft had already approached other companies to withdraw from activities that supported Java™ developments (e.g. Netscape and Intel). By late spring of 1996, senior Microsoft executives were deeply worried about the potential of Sun's Java technologies to diminish the applications barrier to entry [28].

In March 1996, Sun and Microsoft signed a Technology License and Distribution Agreement (TLDA) for the use of Java. The agreement included the incorporation of Sun's Java™ Technology in Microsoft's Internet Explorer 4.0. Late 1996, Microsoft released Internet Explorer 3.0. It was a much-improved version. Some reviewers considered it competitive to Netscape Navigator. In order maximize the usage of Internet Explorer, Microsoft decided that the next version would be more tightly integrated into Windows [28]. Moreover, Microsoft was using its Java license to create its own Java development tools and its own Windows-compatible Java runtime environment. It did so in a manner that undermined Java portability and that was incompatible with Sun's Java products. In the same month that Sun started the PAS application, Microsoft distributed its own incompatible Java toolkit. When Sun applied as a PAS submitter for the second time, it was preparing a lawsuit against Microsoft for copyright infringement. For Sun, the rumors of Microsoft's previous dealings with other players and a premonition of Microsoft's strategy to develop a Windows-dependent Java browser and toolkit would have been reasons not to overestimate its own position in the market. In this market, the step towards international standardization may well have served the purpose of rallying support for Java™. Sun most likely assessed that

its footing in the Java market was not secure enough, which explains its willingness to standardize. On the other hand, it also explains why Sun could not relinquish control over Java.

## 5.2. Withdrawal

*Stated reasons.* Sun withdrew from the PAS process because it did not agree with changes in the PAS procedure decided on in November 1998 [19]. The old procedures still applied, but Sun's status as a PAS submitter would have to be reconfirmed in November 1999, at which time the new rules would apply. The new procedures, according to Sun, implied that Sun would have had to turn standards maintenance and the evolution of Java over to JTC1. Moreover, standards maintenance would not be restricted to minor adjustments such as bug fixing. JTC1, on the other hand, remarked that the changes were clarifications [29].

Comparing the 1999 version of the PAS procedure with the previous version (1995), in the latter version handling of standards maintenance is settled 'in accordance with the agreements made between JTC1 and the recognized PAS Submitter'. The 1999 version stipulates that the normal JTC1 rules for maintenance apply, regardless of the origin of the International Standard. JTC1 would take the lead in corrections to defects and - which will have alarmed Sun - revisions of existing standards. Reacting to Sun's objections, the JTC1 chairman writes, that "the clause addressing the topic of maintenance in the revised JTC 1 PAS procedure is consistent with the comments made by a number of JTC 1 National Bodies that voted to approve Sun as a PAS Submitter but noted the need for JTC 1 involvement in the maintenance of the resulting International Standard." [30]

But much had happened behind the scenes. Sun attributed the changes made to the PAS procedure to lobbying by Microsoft, Hewlett-Packard (HP) and others from the 'Wintel world' [31]. (Microsoft wanted its own Java functionality's enabled.) Sun withdrew because it felt that the change of procedures was only a next stage in the opposition. The procedural changes signaled that Sun would encounter problems when submitting the Java specification. For example, a Java Study Group had been installed in JTC1 Sub-Committee 22 (SC22) and people were discussing how they were going to change the Java specification. It was at that point that Sun seriously started considering alternatives.

*Interpretation.* Sun judged that JTC1 would probably not agree to ratify Sun's work in view of the influence of the 'Wintel-world in JTC1. But, apart from the reasons Sun gave for withdrawing, there were developments in the market that threatened Sun's position, occurrences which increased Sun's desire to keep a grip on Java

developments. Firstly, Microsoft did not abide to the Java licensing agreement, and posed a threat to cross-platform compatibility. In October 1997, Sun filed a complaint against Microsoft for copyright infringement. In March 1998, the court granted Sun's request for a preliminary injunction. Microsoft was not allowed to use the Java Compatible trademark unless its products passed Sun's test suites. In May, Sun filed a complaint for unfair competition. In November 1998, the court ordered Microsoft to change its software and development tools. Microsoft appealed against the ruling [10].

Secondly, in the same period there were disquieting developments in the area of real-time embedded Java. Hewlett-Packard (HP) announced in March 1998 that it had developed a *clean-room* version of real-time embedded Java, that is, a version that was developed without looking at Sun's source code (Concerned is a manner of reverse engineering by which Sun's IPRs on Java are circumvented.). In June, the US National Institute for Standards and Technology (NIST) started organizing workshops to develop specification requirements for real-time Java. Sun participated, as did competitors such as HP and Microsoft [32]. In November 1998, a Real-Time Java Working Group (RTJWG) led by Microsoft and HP was formed. Sun did not participate. The RTJWG approached the US national standards channels, that is, the National Committee for Information Technology Standardization (NCITS/ NIST), to formalize its standards work. But in January 1999 its request was turned down because NCITS feared this could lead to fragmentation of the Java market. The RTJWG subsequently founded the J Consortium. Meanwhile the Real-Time Expert Group (RTEG) was formed within the Java Community Process, a group that was led by IBM.

The RTJWG activities were disquieting to Sun, because real-time Java draws on the base specifications of Java™. According to the experts whom Sun consulted, it was not possible to write real-time specs in a useful way without making changes to the base specifications. There was therefore a risk that competitive developments in the field of real-time Java would affect the work done on Java™ within Sun's JCP.

Sun reacted to the market pressure and to changes in the PAS procedure by elaborating the procedures for Sun-led Java community participation, withdrawing from JTC1, and exploring alternative options for international standardization. In December 1998, Sun issued its first version of the JCP and presented its Community Source licensing model (see section 3.2). They were designed to signal that Sun had taken the criticism of 'benevolent dictatorship' to heart and accepted more far-going influence of the community on Java development. The Community Source model, which partly sympathized with the open source movement, was to underscore Sun's new approach. The new approach mainly served to re-orient

players in the field of real-time Java. Sun's JTC1 initiative had failed to keep the real-time Java dissidents in line. The withdrawal in itself was based on Sun's assessment that it would not be able to maneuver the Java specification through the PAS procedure unscathed. It was a move that followed from its compatibility control strategy. 'Re-orientation of the market' was not at stake, because those involved with the Java™ programming environment publicly heard about Sun's withdrawal when Sun had already approached ECMA (May 1999). To them, Sun was still pursuing the standardization path.

## 6. ECMA, the second attempt

In April 1999, Sun formally approached the ECMA to discuss Java standardization [33]. Sun initially proposed that ECMA would carry out 'passive maintenance' of the Java standard, meaning that Sun's JCP would still determine Java development [36]. But ECMA refused to endorse this approach. The two parties ultimately agreed to the installment of a technical committee on Platform-Independent Computing Environments (TC41) which would 'standardize the syntax and semantics of both general-purpose and domain specific platform-independent computing environments.' The committee would develop a standard for a cross-platform computing environment based upon the Java 2™ Standard Edition Version 1.2.2, a specification that consists of the Java Language Specification, the Java Virtual Machine Specification, and the Java API Core Class Library Specification. The aim was to contribute the standard to ISO/IEC JTC1 by means of the Fast Track process. The ECMA General Assembly gave its approval in June 1999.

The first TC41 meeting took place in October 1999. It was chaired by IBM. During the meeting, Sun emphasized that the TC should focus on 'edition rather than addition' of the Java specifications. Sun provided the main editor. The JTC1 SC 22 Java Study Group, with which the ECMA liaised, would be asked for input before formally invoking the Fast Track process. Three task groups were installed to tackle the work. A Microsoft representative chaired the group working on the API specifications. Sun was to distribute the Java 1.2.2 specification on CD-ROM at the meeting. However, at the end of the two-day meeting a Sun representative announced that Sun lawyers required more time to consider the IPR issues involved [34]. The second meeting was set in January 2000.

In December 1999, Sun made public that it would not contribute the Java specifications to ECMA. At the January meeting, the TC41 participants debated whether it would be feasible to draft a Java standard without Sun's contribution. But some large companies objected (Fujitsu, Siemens, HP and Compaq). In March 2000 the TC was disbanded.

### 6.1. Initiative

*Stated reasons.* Sun chose ECMA because ECMA had close ties with the formal European and international standards bodies and an A-liaison with JTC1, which gave it access to the Fast Track procedure. Sun understood that in the past ECMA standards had been submitted to a yes/no vote in JTC1 without any modifications, and often successfully so. If Java would become an international standard, customers, partners and developers would feel more confident about investing in it [35]. But, Sun said, it would also be pleased if Java would remain an ECMA standard [31].

From Sun's standpoint, ECMA TC41 would edit the Java version that resulted from Sun's JCP trajectory, because there were products based on it and there was a developer community working to the specification. Sun was under the impression that ECMA had agreed that Sun would retain copyright of the specifications during the standards process, and that ECMA would copyright the resulting standard. The latter was necessary to submit it to JTC1 through the Fast Track procedure. (Although Sun would not claim copyright of the standard, it would hold on to IPRs such as the Java name and the Java Compatibility logo, which had a business value to Sun.)

Furthermore, TC41's program of work was specifically limited to the Java Standard Edition version 1.2.2. Any risks, which Sun was taking, would be restricted to this Java version. More far-reaching changes would be part of a new Java version, a development process that would take place within the JCP environment [36].

*Interpretation.* ECMA was an open standards consortium and thus an answer to continuous pressure from licensees and real-time Java developers to open up the Java development process. Many large companies were members. So ECMA processes also promised to be relevant in respect to 'market coordination'. Sun's move further suggested consistency in its aim towards international standardization. But at the same time, the move was an alibi for withdrawing from the PAS procedure without gravely letting down those who were pressing Sun for open standardization.

Sun's position in ECMA was stronger than in JTC1. Sun participated at the time in the ECMA Coordinating Committee (Mr.R. Cargill) and shortly after in its Management (Ms.V. Horsnell, treasurer); and the acting chair of the JTC1 SC22 Java Study Group, with which ECMA liaised, was a Sun representative (Mr. J. Hill). Sun further controlled the conditions under which the process would take place by means of its IPRs and by restricting the scope of the program of work. Perhaps, too, in the preparatory period of defining TC41's program of work, Sun had less reason to fear Microsoft. The judicial system

was partly checking Microsoft's undermining actions with regard to Java compatibility.

In the set up of this standards initiative, Sun had a more focused control strategy than during the PAS initiative. Its emphasis appears to have been on content-oriented standardization.

### 6.2. Withdrawal

*Stated reasons.* Sun's official reason to withdraw from the ECMA process was that "(...) ECMA has formal rules governing patent protections; however, at this time there are no formal protections for copyrights or other intellectual property." [37] Unofficial Sun sources indicated that problems had arisen between the ECMA GA meeting (June 1999) and the first ECMA TC41 meeting (October 1999). These concerned the timing and place of the first meeting, which was scheduled months later than Sun had intended, and procedural issues. (Certain companies insisted that the committee would not be chaired by Sun, that the editors would not be Sun people, and proposed that Microsoft coordinate the development of API specifications.) There were also hints, according to Sun, that the oral agreement on copyright, as Sun understood it, would not be upheld. Sun became wary.

At the first committee meeting, Sun lawyers were taken by surprise by the ECMA secretary general's explanation of IPR rules regarding contributions to standardization. As a rule ECMA documents were not copyrighted. Regarding the copyright status of the Java specs, Sun's contribution would become an ECMA document when it was assigned a TC document submission number. When Sun representatives protested, the ECMA secretary general proposed to explore means by which Sun could maintain copyright during the standards process. ("Contributions from member companies to ECMA can be copyrighted, and can retain their copyright status if the owner of such a specification allows ECMA to freely use the contents of the contribution for the development of an ECMA Standard." [38])

The problem was, firstly, that the parties (Sun and ECMA) had a different view on what was previously agreed, and in particular who was to copyright the Java specs during the standards process. But, secondly, Sun's ideas with respect to the meaning of copyright at that point appeared to differ from ECMA's. Sun differentiated between a copyrighted specification and a copyright of the contents of the specification (i.e. roughly speaking, the difference between paper and software). The problematic part was how TC41 would handle the latter copyright interpretation, which was new to all concerned. At the subsequent meeting of the ECMA Coordinating Committee (November 1999), Sun explained the distinction, and said that it intended "to provide ECMA with a derivative copyright but that this has to be treated as an IPR, under a copyright license agreement" [39]. The conditions of such an agreement were not yet decided on. Early December, Sun announced its withdrawal.

George Paolini, vice president of Java community development at Sun, provided another reason for Sun's withdrawal. He said in a letter to ECMA that Sun had decided to keep control of Java within its Java Community Process. "The Java Community Process has expanded its level of activity to a point where we now believe the interests of the entire Java community will be best met by continuing to evolve the Java specifications with the open JCP process." [16] By then, a proposal for the second version of the Java Community Process had been developed.

*Interpretation.* The events that took place before the first ECMA TC41 meeting, indicated that Sun's influence on the standards process was under attack: procedural issues were discussed that would undermine Sun's position. Furthermore, according to a member of the ECMA Coordinating Committee the prior informal agreement about copyright issues was ambiguous.

The steps which Sun took in the months following its withdrawal give credence to Sun's official reason to withdraw. The industry association of European Information and Communication Technology Industry Association, founded in January 2000, installed a Standards Policy Group chaired by Sun. The policy group

**Table 3. Summary of the findings**

Sun actions > Coordination strategies	Formal Standardization (JTC1)		Consortium Standardization (ECMA)	
	<i>Initiation</i>	<i>Withdrawal</i>	<i>Initiation</i>	<i>Withdrawal</i>
Technology-oriented compatibility control	-	X Java would not survive PAS unscathed	x Focus on Java version 1.2.2	X Procedural disputes Copyright ambiguity Sun-Microsoft lawsuit



Orchestration of market orientation	X Heighten market's commitment to Java	x JCP installed to attract real-time Java developers	X Heighten market's commitment to Java	-
-------------------------------------	---	---	---	---

was to develop a position on the licensing terms of software technology embedded in standards protected by copyrights rather than patents [40]. Sun also planned to raise the issue at a meeting of the European ICT Standards Board, but refrained from doing so before the meeting [41]. Lastly, Sun called together a Standards IPR Forum meeting during the Open Group Conference (April 2000, London) to address, among other things, ownership of copyright on submissions.

However, the primary issue was not that the copyright agreement was ambiguous and informally arranged - probably both ECMA and Sun initially had an interest in this arrangement. The above-mentioned procedural disputes between June (approval of the TC41 work program) and October 1999 (the first TC41 meeting) seem crucial. Moreover, in August, Sun heard that in its ongoing lawsuit against Microsoft the court had granted Microsoft's appeal against the preliminary injunction for copyright infringement. The appeal was, in brief, that the punishment did not fit the crime committed (i.e. a breach of contract should not be punished by means of an injunction). This verdict was a blow to Sun, and had consequences for Sun's stance in ECMA. If Sun would loosen its IPR claims for the purpose of ECMA standardization, it might jeopardize its position in the next stage of the lawsuit.<sup>7</sup> Furthermore, I could also imagine that the verdict raised Sun's doubts about what legal protection a copyright offers (-although this was to my opinion not the issue in the August trial). This would explain Sun's introduction of a dual meaning of copyright. In sum, the procedural issues and the Sun v. Microsoft lawsuit fuelled Sun's wariness. By not clearing the copyright issue beforehand, Sun could introduce a new meaning of copyright, one which would not be acceptable to the ECMA TC, to pave the way for total withdrawal. "[Sun] just does not want to give up control", as the ECMA Secretary General, Jan van den Beld, told the press [43], and it had several reasons not to do so. Possibly Sun did not believe Java was stable enough or had achieved sufficient critical mass to relinquish control [43]. Whatever reason presided with regard to ECMA standardization, Sun's actions focused on preserving control over the Java™ specifications.

<sup>7</sup> Informal communication with ECMA TC41 participants. The verdict was confirmed in January 2000. Sun's compliant against Microsoft for unfair competition was granted. [42]

## 7. Conclusion

Sun primarily initiated standardization in JTC1 and ECMA because an international standard implied stability, would increase market confidence and would therefore encourage commitment to Java. It wanted JTC1 and ECMA to 'ratify' the existing Java™ specification and did not seek the involvement of their members in its development. Rather, it sought commitment from the clients of these standards bodies (i.e. implementers of JTC1 standards). That is, Sun's motives were not technically but market oriented. See Table 3.

It withdrew from JTC1 because it suspected standards politics behind procedural changes, because of incompatible and competing market developments, and - above all - because it expected that its Java specification would not survive the PAS procedure unscathed. Sun intensified its compatibility control strategy in subsequent negotiations with ECMA. To minimize risks, it focused its standards initiative on a specific version of the Java specifications. However, the procedural disputes that preceded the first ECMA committee meeting made Sun wary. Added to new developments in the lawsuit with Microsoft, Sun referred to ECMA's ambiguous copyright rules to pull back from ECMA standardization. Concerns about the technical coordination of Java developments dominated during this period.

Sun pursued a protective and defensive control strategy. Whether it should instead have followed a more offensive strategy, based on confidence in a market-coordinated development of platform-independent Java, is a matter for debate. However, where a key technology such as Java™ is concerned, the stakes are high. The case suggests that in these situations formalization of a proprietary de facto standards is unlikely to be successful.

## References

- [1] M.B.H. Weiss & M. Sirbu, "Technological choice in voluntary standards committees: an empirical analysis", *Economics of Innovation and New Technology*, 1, 1990, pp.111-133.
- [2] P. Grindley, *Standards, Strategy and Policy: Cases and Stories*, Oxford University Press, Oxford, 1995.
- [3] C.F. Cargill, *Information Technology Standardization. Theory, Process and Organizations*. Digital Press, Digital Equipment Corporation, 1989.
- [4] M.J. Bonino & M.B. Spring, "Standards as change agents in the information technology market", *Computer Standards & Interfaces*, 12, 1991, pp.97-107.

- [5] P.A. David & S. Greenstein, "The economics of compatibility standards: an introduction to recent research", *Economics of Innovation and New Technologies*, Vol. 1, 1990, pp.3-41.
- [6] H. de Vries, "Formele ICT-normalisatie: passe?", *I&I*, 17, 1999, pp. 28-35.
- [7] S.K. Schmidt & R. Werle, *Coordinating Technology. Studies in the International Standardization of Telecommunications*, MIT Press, Cambridge, Mass., 1998.
- [8] J. Farrell & G. Saloner, "Installed base and compatibility: innovation, product preannouncements and predation" *American Economic Review*, 76, 1986, pp.940-955.
- [9] F. Vercoulen & M. van Wegberg, "Het spel van samenwerking en concurrentie: Hybride standaardselectieprocessen in de ICT sector", *I&I*, 17, 1999, pp. 19-27.
- [10] T.M. Egyedi, *Compatibility strategies in licensing, open source software and formal standards: Externalities of Java 'standardization'*, Contribution to 5th Helsinki Workshop on Standardization and Networks, 13-14 August 2000.
- [11] Sun, *Amended Complaint*, October 14, 1997.
- [12] R.P. Gabriel and W.N. Joy, *Sun Community Source License' Principles (SCSL)*, 1998.
- [13] Sun, *The Java Community Process (sm) Program Manual: The formal procedures for using Java Specification development process* (version 1.0, December 1998).
- [14] E.R. Harold, *The Java Gated Community Process*, 1999, [http://metalab.unc.edu/javafaq/editorials/jcp.html].
- [15] M. Vizard, "Java Lobby president calls for reform", *InfoWorld Electric*, November 19, 1998. [www.infoworld.com]
- [16] S. Shankland, "Sun offers olive branch to Java partners", *CNET News.com*, March 1, 2000.
- [17] Sun, *The Java Community Process (sm) Program Manual: The formal procedures for using Java Specification development process*, version 2.0, review draft, April 2000.
- [18] Ch. Babcock, "Java: Can Sun control the flood", *Inter@ctive Week*, June 12, 2000, [www.zdnet.com].
- [19] ISO/IEC JTC1, *The Transposition of Publicly Available Specifications into International Standards - A Management Guide* -Revision 1 of JTC 1 N 3582, N 5746, January 1999.
- [20] ISO/IEC JTC1, *JTC1 Directives*, 1999.
- [21] T.M. Egyedi, *Grey fora of standardization: a comparison of JTC 1 and Internet*. KPN, KPN Research, Leidschendam. R&D-RA-94-1235, 1994.
- [22] T. Clark, "Java standard voted down--for now", *CNET News.com*, July 30, 1997.
- [23] Sun, *Sun response to ISO/IEC JTC1 N4811 and N4833*, 1997 [www.Sun.com].
- [24] ISO/IEC JTC1, N5090, November 1997.
- [25] "Sun's ISO application progresses", July 16 1997 [www.sun.com].
- [26] N. Brookes, "Has Sun gone off on one?", contribution to a discussion on *JavaLobby Pro-Sun or Pro-Java?*, 26 January 2000.
- [27] Sun press conference, Question and Answering session, November 1997, [www.sun.com].
- [28] *United States of America v. Microsoft Corporation. Findings of Fact*, United States District Court For The District Of Columbia, Civil Action No. 98-1232 (TPJ), 1999.
- [29] ISO/IEC JTC1, *PAS Transposition: JTC 1 finishes trial, improves its guidelines and moves to regular operation*, 1999-9-8.
- [30] *JTC 1 Chairman's Comments on Recent Press Articles Regarding Sun Microsystems' PAS Submitter Status*, 1999-5-1, [www.jtc1.org].
- [31] S. Shankland, "Sun revising plan for Java recognition", *CNET News.com*, April 29, 1999; 'Sun seeks control of Java process', *CNET News.com*, May 6, 1999, [www.znet.com].
- [32] E.D. Jensen, *Real-Time Java Status Report*, Revised Draft, January 24, 1999, [http://www.real-time.org/].
- [33] S. Shankland, "Sun renews Java standards effort", *CNET News.com*, May 5, 1999.
- [34] ECMA, *Minutes of the 1st meeting of ECMA TC41 held in Menlo Park (USA) on 25th - 26th October 1999*, ECMA/TC41/99/16, 1999.
- [35] J.C. Perez, "Sun changes strategy for Java's ISO approval", *IDG News Service*, May 6, 1999.
- [36] C. Sliwa, "Java standard switch: Will new process ease cross-platform compatibility?", *Computerworld*, May 7, 1999.
- [37] Sun press release, December 7, 1999.
- [38] ECMA CC 11-12 November 1999.
- [39] ECMA, *Minutes of the meeting of the coordinating Committee held in Geneva on 11th - 12th November 1999*, ECMA/GA/99/127.
- [40] The EICTA position paper is referred to in *CSN 337-03 Sun's Up To Something*.
- [41] *Software Technology Embedded in or Becoming a Standard*, ICTSB18 (00) 22, discussion paper.
- [42] *Sun Microsystems v. Microsoft Corporation*, United States District Court For The Northern District Of California, January 24, 2000, No. C 97-20884 Rmw (Pvt), *Order Re Sun's Motion To Reinstate November 17, 1998 Preliminary Injunction Under 17 U.S.C. § 502; Preliminary Injunction Under Cal.Bus. & Prof. Code §§ 17200 Et Seq.*
- [43] J. Nicolai & L. Rohde, "Sun confers with licensees over control of Java", *IDG News service*, 3/3/2000.