1

Information and Communications Technology and *Shiso*
— *Shiso* as a Capability for Science and Technology —

Susumu Hayashi
Affiliated Fellow

1 Introduction

When the integration of sciences and the humanities refers to collaboration between science/engineering and the humanities/social sciences, it constitutes an essential element of science and technology policy, taking how science and technology are positioned in modern society into consideration. Japan's Third Science and Technology Basic Plan[1] repeatedly emphasizes the importance of this issue, especially in the environmental field.

However, this article views relations between science and technology and the humanities and social sciences in a way that is completely different from the notion of integration as suggested above. The article points out how these relations can be a key factor in determining the growth or decline of Japan's information and communications sector, especially the software sector, and proposes the need for a science and technology policy incorporating this perspective.

This article is an extension of the view proposed in "The Two Rationalities and Japan's Software Engineering"[2], a feature article in the September 2004 issue of "Science & Technology Trends" that analyzed the weakness of Japan's software sector from the perspective of rationality. While the earlier paper focused on software technology, the present paper encompasses the information and communications field as a whole, and portrays social systems for researchers as the source of science and technology. This perspective is what I refer to as *shiso* (see definition below), which is a Japanese term, and the issue is not confined to information technology alone. As most science and technology fields, and even the entire society, are becoming computerized and cybernated*1, the need is growing for science and technology policies that incorporate *shiso*, particularly policies aimed at the development of human resources.

1-1 This article's definition of *shiso*

This article provides a discussion intended to contribute to the promotion of Japan's science and technology. From this perspective, the usual meaning of the Japanese term *shiso*-a social or political thinking structure-is not appropriate for the discussion. *Shiso*, as used in this article, is dissimilar to the one used to explain Kantian philosophy and Marxism**, but is rather close to the meaning of "philosophy" as in "the practical philosophy of the Toyota Production System." To avoid confusion, what this article refers to as *shiso* is first defined, and then followed by a discussion on the need for *shiso*.

The Kojien Japanese dictionary defines *shiso* as follows: (1) thought; (2) <philosophy> (a) results of contemplation not just through intuition prior to any judgment, but through such intuition combined with logical reflection; the content of such thinking, especially structured thinking; (b) a system of comprehensive ideas on society and life, often with social or political implications.

Placing emphasis on the systemic aspect mentioned in (2)-(a) and the comprehensiveness in (2)-(b), this article defines *shiso* as shown in Table 1.

According to the definition in Table 1, *shiso* encompasses everything that arises from spirit (as in the pioneering spirit, the explorer's spirit, and the frontier spirit) and philosophy (the philosophy underlying Japanese-style manufacturing), to culture (corporate culture, Toyota's culture), soul (an engineer's soul), and doctrines and "isms" (Taylorism, Fordism, rationalism). Because this article broadens the meaning of *shiso* so much from ordinary usage, even technical words are included. According to this definition, "The Protestant Ethic and the Spirit of Capitalism," a work by sociologist Max Weber, may be regarded as a treatise explaining the cause-effect relation between two different *shiso*s such as this: "the Protestant *shiso* produced the *shiso* of modern capitalism."

2 *Shiso* in information and communications technology: Software engineering and the development of the Internet

This chapter explains what *shiso*, as defined above, actually represents, examining three notable cases found in the history of information and communications technology. These examples demonstrate two things: (1) *Shiso* in science and technology has different phases; (2) *Shiso* has had strong positive and negative impacts on the history of the U.S.-led development of information and communications technology.

The first case helps explain the role of *shiso* in recent trends in software production technology. This is an attempt to address *shiso* as a useful methodology for systems and systems development processes, and is an extension of previous articles by this author and his fellow researchers[2,3]. *Shiso* has two meanings in this case. One is close to *shiso* as in "this system's design *shiso*," which is a concept specific to each software development project and lasts for a relatively short time on a small scale. *Shiso* here can be defined as "lower level" *shiso* because it is more specific and individualized. The other meaning is related to the attitude where "each development project shall have a lower level *shiso* specific to it," and this will be used to point out how this attitude has been advocated as a new paradigm for software production by an increasing number of researchers since the late 1990s. This represents a major shift in the software development process paradigm, and is a "higher level" *shiso* as suggested in the remaining two cases, because it is a common idea among different development projects over the long term.

Table 1 : This article's definition of "*shiso*"

Shiso refers to a specific pattern of thinking being shared, and handed down from one generation to the next, in a specific group (e.g., a religious order, professional group, ethnic group, company, university, community), and a system (a set) of such patterns. However, shiso is unlike instinct and custom in that any action taken according to shiso must be conscious, where "conscious action" means the person taking that action is aware that it follows a distinctive pattern.

This leads to the second case, which explains a higher level *shiso*. The second case will reveal the role that *shiso* played in the development of the Internet, the largest direct effect exerted by information technology on society. The last case is Web2.0, a typical case of higher level *shiso* and a major recent trend in information technology. This subject is analyzed from the viewpoint of *shiso* as defined in this article. Web2.0 is not a collective term referring to a set of new technologies, but a clear embodiment of *shiso*. It has something in common with the other two examples and is strongly linked to current social changes.

2-1 Software development in relation to theoretical and *shiso*-oriented skills

Current trends in software engineering mean that software engineers have had to start acquiring capabilities that may be called "theoretical skills" and "*shiso*-oriented skills." A. Cockburn, a prominent American software consultant, described in his recent book[4] how the theory propounded by the 2006 Turing Award winner P. Naur, known as "programming as theory building," is in fact practical knowledge for software developers. What Naur calls theory is not a set of printable rules, such as programs and specifications, but a more comprehensive idea consisting of knowledge possessed by those who create rules (programmers), especially that on how to create rules, and the process for creating and maintaining rules. Naur's theory uses the terminology of British philosopher G. Ryle, who was influenced by the ideas of Wittgenstein, one of the most distinguished philosophers of the

20th century. The above study by Cockburn[4] also cites design scholar P. Aine's explanation of software development based on Wittgenstein's language theories, suggesting that Cockburn views practical knowledge on software engineering as an extension of the philosophy (*shiso*) of Wittgenstein. What Naur means by theory building is the activity of matching elements in real-world activities with formal and symbolic operations on the computer. From this perspective, Naur has emphasized that the "exercise of theory building" is essential for the education of programmers.

*Shiso*s similar to Cockburn's and Naur's constitute a dominant trend in recent software engineering. For example, the same tendency is clear in "Problem Frames"[5], the latest theory of M. Jackson, the inventor of the Jackson method*2, which some call the world's first program development method. D. D'Souza, who is known for his component-based development method*3, has also stressed this trend for several years and is trying to construct an original *shiso* for software development[6].

These *shiso*s should be considered as skills because of their contribution to improving software productivity. They are "lower level *shiso*s" if expressed by the terms introduced at the beginning of this chapter, or they are the opposite of scientific theories if put with the term "theory" as defined by Ryle and Naur. Modern scientific theories owe much to Western philosophy. The philosophies of Descartes and Leibniz and the *shiso* of Newton have had an impact on the generation of each scientific theory, whether positively or negatively. *Shiso* helped scientific theories to be incorporated into culture and society, consequently allowing science and technology to bring change to *shiso*. This fact suggests that *shiso* is precisely the right tool to harmonize the development of systems that exist as independent theories. That is, what Cockburn, Jackson and D'Souza proposed are, in fact, *shiso*s.

This explanation clearly holds true for agile methods, the latest software methodology that has quickly come into widespread use since the beginning of the 2000s (see Reference[2]). Unlike conventional software development methods, such as the Jackson method, agile methods have been promoted through a group called the Agile Alliance, in which anyone who agrees with the Agile manifesto, a set of values on software development, can participate. In short, what defines agile methods are these values, or *shiso*. Agile methods are based on the *shiso* that the key factor of success or failure in a development project is the relationships among programmers within the development team. This is natural, considering that what Ryle and Naur call theory resides not just on paper, but in the minds of the people involved. Since people act as devices to create and store theories, the state of the mind of each device, such as the courage to accept change and the humility to facilitate ease of communication among team members, influences productivity. Agile methods emphasize such attitudes of programmers, and argue that having a good mind is an indispensable skills of being a programmer. This is very close to what Japanese companies have traditionally promoted as corporate cultures through kaizen activities. In fact, as explained in the following chapter, Toyota and other Japanese-style production and management *shiso*s have influenced the *shiso* of agile methods. This is proof that agile methods may be called a *shiso*.

In summary, the latest software engineering consists of theoretical skills (skills called "modeling") to create systems as individual theories and *shiso*-oriented skills, which refer to structured terms and phrases to enable theoretical skills to be communicated and learned.

2-2 *Shiso* in the history of Internet development

The previous section explained what *shiso* is, taking the latest trends in software engineering as examples. The next section describes examples taken from the early days of modern information and communications technology. *Shiso* played an essential role in the development of the Internet, one of the greatest inventions of the 20th century. Internet technology was born, led by one person's *shiso*, at a time when few people could recognize its potential or visualize its complete form.

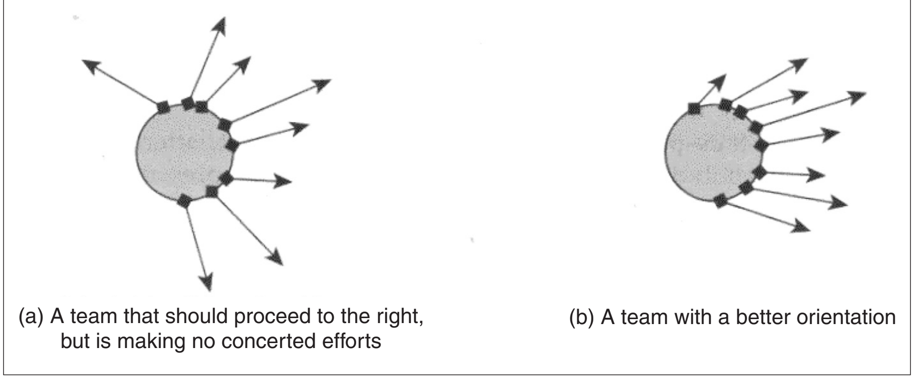
Waldrop[7] and Kita[8] wrote that the *shiso* of a psychologist named J.C.R. Licklider played a critical role in the early stages of the development of the Internet. In particular, Kita's "Internet no *shiso* shi" ("The History of Ideas of the Internet") carries the word "*shiso*" in the title. This book portrays the history of Internet development as a history of ideas or a history of the evolution of technical ideas.

Although personal computers connected through a network constitute the popular image of modern IT society, this was not the dominant technical image in the 1960s and 1970s, when such networked computers had yet to emerge. In the information and communications world before the Internet, research groups dispersed across the U.S. were separately conducting outstanding studies that would go down in scientific history. They were striving toward different goals, while competing with each other. Some were faced with the limitations of thinking inherent to technology professionals, and others suffered from their inability to transcend existing technologies and a lack of evaluation from the users' point of view.

However, technologies resulting from these efforts were integrated by *shiso*s proposed by the psychologist Licklider between the late 1950s and the early 1960s, such as "man-machine symbiosis" and "a network of thinking centers," and were made manifest under the guidance of these *shiso*s. Licklider had once been a development team member of the SAGE system[*4], an epoch-making system in the annals of computer development history. This experience, combined with his nonexpert status in the information technology field, made him a scientist who was capable of roughly estimating the potential of technologies from the viewpoint of a user without technical bias. Coincidentally, this scientist was offered an important post in the U.S. Advanced Research Projects Agency (ARPA), which gave him access to ample funds for use at his discretion. Licklider made the bold decision to spend the funds on emerging elemental technologies corresponding to his *shiso*.

The Roman philosopher Seneca remarked, "If one does not know to which port one is sailing, no wind is favorable." This is sometimes quoted with reference to requirement engineering, a branch of software engineering, as a warning to software development teams who may fail in their endeavors if they do not successfully identify their goals (requirements). Licklider was able to show the direction of the port, or the goal (developing the Internet), through his own *shiso*.

When the subject is an integrated technology like the Internet, the commitment of many excellent research groups is not enough, because their progress will be offset if individual groups are aiming at different goals according only to their own *shiso*s, with no shared *shiso* to align them. This is like a boat whose crew is rowing toward different destinations. By contrast, even small driving forces can produce a great effect when aligned and combined (Figure 1). When the situation was similar to (a) in Figure 1, Licklider used research funds, an element of the research environment, to roughly set a direction (destination port) for researchers to pursue, in a manner that few of them recognized. The result was the launch of the Internet. Once the Internet emerged, the societal need for such technologies as email and the Web became the "port," rapidly



(a) A team that should proceed to the right, but is making no concerted efforts

(b) A team with a better orientation

Figure 1 : Cockburn's explanation of development project directions

Prepared by the NISTEP based on Figures 3-17 and 18 in Reference[4] spreading the Internet in a way few experts had imagined.

2-3 Web2.0 and the open shiso

The last case is Web2.0, referred to by some as "the latest version" of the Internet, and which is derived from Licklider's *shiso*. Web2.0 is an idea proposed in the autumn of 2005 by T. O'Reilly, a well-known author of software-related literature. Umeda introduced the term to Japan in his book[9]. One may still wonder what kind of technology this is. Although Web2.0 obviously has something to do with the Google search engine, even O'Reilly's article[10] does not provide an explicit explanation of Web2.0 technology. Instead, he just lists, alongside common and well known technologies like "Napster," "Wikipedia," "blogs" and "web services," such notions as "syndication" and "participation," which may be regarded as a policy or an attitude.

Like agile software development, which is an alliance of people sharing the same *shiso*, Web2.0 is actually a group of information and communications technologies and services based on a common *shiso*. That is, as agile software development is essentially a *shiso* symbolized by the Agile manifesto, Web2.0 may be expressed as nothing but the *shiso* of Web2.0. This means that an easy explanation of the essence of this *shiso* should not necessarily be sought in software-related literature. As far as I know, the closest idea to the essence of the Web 2.0 *shiso* has been represented by T. Friedman, an American journalist, as the "flattening of the world," which he says is a "new version" of globalization[11]. Web2.0 is this flattening phenomenon in the information technology sector, and is one of the most significant technical factors behind the overall global flattening.

To put it simply, the *shiso* of Web2.0 may be considered as one that aims to transform a society into an aggregated intelligence acting like a huge cyborg, by connecting people's individual intelligence (assumed as CPUs) through information and communications technology. Under this assumption, the performance of the resulting device is dependent on the performance of individual CPUs, or humans, and of the society to which these people belong. This suggests that improving the quality and accuracy of such "information devices" as Google and Wikipedia is not a matter of science and technology, as has been conventionally assumed, but a matter of social and educational policies. The quality of the Wikipedia free encyclopedia varies depending on the language used. The quality of the Japanese language version is generally not as good as the English version. This can be attributed to a disparity in performance between the two language-speaking groups, or more specifically, the size of the Japanese-speaking population and the English-speaking population and the total quality of each group. This disparity in performance is highly likely to be reproduced on larger scales because Wikipedia, Google and similar technologies are now being widely adopted in education as a matter of course, and the Japanese have a low competence in the English language.

It is also obvious that such a social scientific analysis of societies is indispensable for forecasting trends in, for example, the long tail phenomenon[9], which is explained as a "structural change in commerce" as a result of applying the above devices to commerce. Because this kind of issue needs to be discussed in light of not only economic principles, but also from language and cultural perspectives (where the issue is pertinent to numerous countries) and the values in the society.

What is more noteworthy is that the *shiso*s of Web2.0 and flattening are not isolated or exceptional examples. The primary element underlying them is the same as the one behind the open source movement in the software sector. That is, the *shiso* that is optimal for solving a problem concerning an entity that is too vast and too fast-changing for anyone to theoretically predict the future within a reasonable timeframe, is to use social collaboration. That is to say, the best approach is first disclosing as many constraints as possible, then sharing a provisional solution with others through a network of many independent people who have a common goal and *shiso*, and allowing them to refine the solution step by step through a succession of methodical modifications.

Apply this *shiso* to knowledge searching, then the answer is Google; apply it to encyclopedia making, and the result is Wikipedia; and if you apply it to software development, then you will be led to Linux and other open source products. When the same *shiso* is adopted by a custom software development team on a limited basis, the product is agile methods. In the case of Internet development described in the previous section, merely sharing a goal was shown to be insufficient, but the *shiso* of Licklider, who was responsible for allocating the research funds, was able to compensate for this deficiency. "Collective knowledge," a term that has recently gained some currency, is also closely related to this *shiso*. In short, these are all methodologies that aim to transform a society into a production device with unparalleled capacity by ensuring that its members have as much freedom as possible and by standardizing and leveling the quality of these members through education. This device is capable of autonomous reproduction on an enlarged scale, as existing knowledge generates new knowledge incrementally. Since Web2.0 is in essence a *shiso*, things similar to it will appear again even after the term itself falls into disuse, and they will drastically reshape society and science and technology.

3 Negative *shiso* and Japan's concern

The previous chapter presented three cases to demonstrate that *shiso* has played a critical role in the information and communications sector since its infancy through until today, and that this trend is accelerating. However, *shiso* does not always have positive effects. This chapter describes some of the negative impacts *shiso* has had on science, and discusses potential negative effects that the absence of *shiso* in Japan's information and communications technology might have on the country's collective science and technology endeavors.

3-1 Negative *shiso* and its implications

An examination of the history of science shows that *shiso* has significantly influenced science, as mentioned earlier. Many scientists will concur with this view. However, the influence may not have been always positive. In Section 2-1, *shiso* was defined as something inducing theory development. Scientific theory is a kind of theory. If a theory is assumed to represent one field of technology, *shiso* can be considered as a metascience. A science governed by a metascience is naturally dependent on that metascience. Although the examples presented so far are all successful ones, an error in metascience can sometimes exert a negative influence on science.

For example, Michio Kobayashi, an internationally renowned philosopher with a special interest in Descartes, describes the adverse effect of Cartesian philosophy on Cartesian physics[12] as follows. From the point of view of today's researchers, natural science and philosophy were unified in the thinking of Descartes, who was a natural scientist and philosopher, and his philosophy justified his methodologies in natural science. Despite having mathematical methodologies capable of correctly solving the equivalent pendulum problem (finding a pendulum whose movement is equivalent to the swing of a board pivoted at a point), Descartes concluded that the problem was unsolvable. The fundamental *shiso* in his cosmology told him that interrelationships between the board and substances surrounding it were too complex to be solved by his mathematics.

The viability of Hilbert's Program*5 was refuted by the incompleteness theorem. Research into the history of mathematics by myself and others[13] has revealed that this program, proposed by the great mathematician, D. Hilbert, was an attempt to mathematically prove the *shiso* of "solvability," or the notion that every mathematical problem can be solved finitely, which occurred to Hilbert when he was still unknown within the mathematical community. In this case, a wrong *shiso* acted as an impetus for the 20th century mathematical *shiso* known as structuralism. This implies that even a wrong *shiso* can sometimes incubate a right *shiso*.

Looking at the history of software engineering, in the 1980s, B. Boehm devised spiral development, a forerunner of agile development, as an antithesis to the waterfall development model, which views software development as a linear process of "requirement specification → design → implementation → validation → installation → maintenance." Nevertheless, the waterfall model persisted for a long time, and is believed to have exerted a significant adverse influence. This is a typical example of negative *shiso* in software engineering, a discipline where *shiso* is a key factor. Interestingly, the year 1986[14], in which Boehm proposed spiral development, a strong impetus to move away from the waterfall model occurred only a year after S. Kline criticized the linear process[15] of development, claiming that there was no rational basis to confine technological development to the linear model of "science → basic research → R&D → design → production → sales → market." The mid-80s also coincide with the time when the Massachusetts Institute of Technology (MIT) started research on Lean Manufacturing, the U.S. version of the Toyota development method that is one of the sources of agile development in software engineering. Sociologically, this represents a social phenomenon in which simple, modern rationalistic *shiso* began disappearing from U.S. engineering communities as they realized the benefits of Japanese-style kaizen.

3-2 Seepage of information technology and the future of Japan's science and technology

If the most fundamental *shiso* for a county's science and technology generates negative effects, a slowdown can occur in every facet of its science and technology endeavors. The departure from the linear *shiso* since the mid-80s in the U.S., described above, indicates that American engineering prior to that time had been under the negative influence of *shiso*. The question here is whether the same problem can happen to Japanese technology. In fact, I have a concern that Japan may suffer from such negative *shiso* effects in the future.

The results of the seventh Delphi survey[16] conducted by the National Institute of Science and Technology Policy (NISTEP) in 2000 provide an interesting perspective: The importance of the information sector, which is regarded as a promising growth technology, will fall sharply in 2010 and beyond. After an additional survey was conducted, the report attributed this result primarily to the view held by many researchers and engineers that technologies in the information and communications field will be assimilated into other fields, becoming nonexistent as an independent area of research. To put this another way, these professionals predict that everything will become computerized and cybernated. This idea resembles the theory of Web2.0, which Umeda explained in his book[9] by using the phrase "that side and this side."

More specifically, many information technologies will exit the field that is still referred to by the outmoded name "information processing" (and is viewed as a special and independent technological field), and will eventually come to assume a position in every engineering field analogous to the one that mathematics occupies today. However, unlike mathematics, information and communications technology is definitely a technology. From this perspective, the above phenomenon may be understood to be a phenomenon in which a technology field named "information" is seeping across the boundaries into other fields. This phenomenon is the cybernation of engineering that many engineers and technologists are experiencing. The trend is likely to continue.

Web2.0 is actually a *shiso* indicating that the cybernation of society will advance without foreseeable limits, encouraging user participation. Probably it is reasonable to expect that this cybernation, along with the flattening, will progress further. If this prediction of the seepage of information and communications technology proves to be right, future science and technology will incorporate more elements derived from software engineering in nature.

However, the Third Science and Technology Basic Plan[1] points out the weakness of the software sector in Japan, as does a survey by the NISTEP[17], which suggests that the sector lags behind the U.S. These views coincide with my arguments in an earlier article[2], which also showed the basis for this assessment. If this weakness of Japan stems from a lack of rationality, or *shiso*, as argued in this earlier article[2], it may end up pervading the industry and even the entire society. Chuma's study on the decline in the competitiveness of Japan's microlithography industry[18] represents a concrete example of this phenomenon. Now that such an example exists, the possibility of a worst case scenario must not be ignored.

The possibility of a worst case scenario is high, in light of the fairly common view that there is no *shiso* in science and technology in Japan. One person warned of the danger of no *shiso* more than 100 years ago: E. Bälz, a German teacher working at Tokyo Teikoku University during the Meiji era. In 1901, in a famous speech[19] marking the 25th anniversary of his service to the university, Bälz expressed serious concern about Japan's science and technology, or more precisely, the future directions of its education. He referred to European science and technology, which Japan was then trying to acquire in order to maintain its sovereignty, as "the great spiritual principle" nurtured by the history of Europe, and compared European science and technology to trees that bear abundant fruit if grown well. Bälz criticized the Japanese people (in the Meiji era) for assuming that science was a machine that could always produce the same products wherever in the world it was installed-an approach that would only bear fruit for the time being-, and encouraged them to learn the spirit, not the technique, of science from visiting international scholars such as himself. Quite a few scientists and engineers would probably agree that Bälz's criticism still holds true for today's Japan, even after a lapse of more than 100 years.

4 Conclusion

Previous chapters discussed the significance of *shiso* in science and technology, but the role of *shiso* varied between the examples shown. However, these examples can be considered as the occurrence of the same phenomenon in different settings, from the perspective that capabilities to manipulate "things without shape or substance" that warrant the name of *shiso* are becoming increasingly important, as the social structure becomes more complex and standardization and informatization progress as ways to bring order to complexity.

Many of these capabilities are of the same type as those falling under the umbrella of the liberal arts. At least in Japan, liberal arts have been regarded as knowledge for character development, rather than for directly contributing to society or productive activities. Consequently, many engineers and scientists, including myself, may have felt somewhat proud of being free of the "useless" *shiso* of the liberal arts.

Under such circumstances, liberal arts education at Japanese universities is deteriorating. Most science students, especially those taking engineering, tend to have little interest in the *shiso* of the liberal arts, and most of the faculty believe that there is little need for such education. However, *shiso* contributes significantly to science and technology on various levels, as the present article has explained. What is more noteworthy is that not only has *shiso* contributed to epoch-making events in the history of science and technology, such as the development of the Internet, but it is becoming an indispensable capability in the field of reliable technology where daily software production takes place. The significance of *shiso* as a methodology in science and technology is growing in many areas. *Shiso* also represents a competitiveness issue relating to the survival of Japanese industry.

Web2.0 trends, as typified by Wikipedia, Google, and long-tail markets, cannot be predicted without social scientific research. More importantly, it is impossible to predict future developments in American information and communications technology circles, which intend to promote Web2.0 more aggressively and maintain the present dominance by understanding Web2.0 as an independent *shiso* and sharing it as a *shiso*.

Attention should be paid to the fact that in the U.S., Google and Web2.0 are discussed by computer science students and engineers rather than by economists. That is to say, in the U.S., new technologies and disciplines are often created by experts in information and communications technology based on *shiso*, in the same way as the Internet was developed. By contrast, such technologies or technology experts are very rare in Japan. The Japanese seem to just keep focusing on producing more accurate components, at a time when the source of competitiveness has shifted to how to design systems for a society acting as production equipment.

If this situation is left unaddressed, Japan will lag farther behind the U.S. in the information and communications sector, and will be forced to continue to trail emerging India and even China. The primary argument of this article is that this problem is mainly dependent on the *shiso* and the type of thinking conducted by scientists and science and technology policy makers.

Since *shiso* is essentially pertinent to human beings, the only possible remedy must be improving education. One way to nurture talent like Licklider is to introduce the double major system*6, like American universities. This method should be effective and relatively easy to introduce into universities with outstanding liberal arts teachers and high-quality science students, although it could backfire if science students with poor liberal arts' skills and understanding were forced to endure uninteresting lectures. Since not all scientists and engineers assume a post like Licklider's, the introduction of the double major system requires that the level and goals of each educational institution need to be considered.

In the example of Web2.0, many scientists and engineers sharing the same information technology *shiso* and culture have been voluntarily collaborating, rather than being led by a science policy or other programs, to pursue a single goal. Their *shiso* is spread through university education, as represented by the fact that Google and Yahoo, two leading Internet companies, were founded by graduate students of Stanford University. The presence and importance of the Palo Alto culture that formed in the area around the university impressed me in the 1980s and 1990s, when I, as a computer scientist, visited Stanford University's Computer Science Department on a regular basis. The ability of Silicon Valley's environment and culture to incubate new information technologies is vividly depicted in Umeda's recent book[9]. Building such an environment is exactly what Bälz described as growing "trees that bear abundant fruit" of science and technology in his speech[19] cited in Section 3-2.

In contrast, Licklider's example shows that if a science and technology policy is led by a person with *shiso* or vision, rather than pursued through collective creation as in the case of Web2.0, it can produce highly valuable results. *Shiso* is not easy to communicate to others. Communicating *shiso* is far more difficult than communicating technology because it is essentially implicit knowledge. As this example suggests, an effective way of successfully implementing a science and technology policy according to *shiso* is to empower a talented researcher with *shiso* or vision to lead the project. This key scientist must have sufficient knowledge and skills as a policy maker, as a policy practitioner, and as a politician. Science history indicates that in Europe and the U.S., there were a number of legendary scientists who served as prime ministers, ministers, politicians or policy practitioners. By contrast, in Japan, the only memorable figure would be Akito Arima, a physicist who was once the Minister of Education, Culture, Sports, Science and Technology. To accelerate the nurturing of such talent, Japan should seriously consider introducing an educational program, such as the double major system, to its principal universities, with the aim of developing students that have balanced abilities in both science and liberal arts.

From my experience as an educator, theoretical skills and associated *shiso*-oriented skills, explained in the argument on software engineering, are easier for science students to understand when their usefulness is taught through practical projects. Although being *shiso*-related, these skills are concrete and practical enough to be acquired through the practice of Design Patterns*7 and UML-based development*8. Universities would be able to teach such small-scale *shiso* by the seemingly paradoxical approach of revising their curricula for information-related departments to emphasize hands-on laboratory activities and practical projects. The result would be university versions of apprenticeships and on-the-job training, which would require faculty who could understand and exploit such *shiso*. Unfortunately, current information-related departments in Japan lack such faculty, and it has been observed by some that Japan has only a few laboratories where software engineering is studied in the true sense. Alternatively, acquiring such human resources from companies is infeasible because few firms have the capacity to develop such talent. On the contrary, companies expect universities to assume such a role.

Realistic solutions under such circumstances are, for example, hands-on laboratory type education led by outstanding international engineers and overseas apprenticeship programs. Likewise, the development of Licklider-like human resources does not have to necessarily take place in Japanese universities; giving students opportunities to learn and research at overseas universities should be considered. There is no doubt that Japan's levels of research and education in individual science and technology fields are as high as foreign ones, but when it comes to nurturing scientists with *shiso* as defined by the present article, especially those who are capable of policy making, current Japanese universities have little experience. As far as this issue is concerned, the knowledge that is worthwhile learning from Western universities, which have a long history in science and technology, has hardly decreased since the days of Bälz. The government should consider the strategy of sending students overseas in an attempt to develop such human resources. In particular, instead of undergraduate and graduate students who have little time to acquire political literacy, a viable approach to nurturing scientists with political insight would be to dispatch young researchers and engineers overseas, and have them engage in policy-making jobs while overseas in order to acquire essential *shiso* skills.

From this perspective, it is worthwhile noting that Umeda, who was the first Japanese to address the theory of Web2.0, and other Japanese living in Silicon Valley, are assimilating *shiso* and implementing projects to give young Japanese the same experience they had while living and working in California. Supporting such activities of non-profit organizations is yet another measure the government can take.

Explanation

** The Japanese term *shiso* is a unique word for which no English word can exactly substitute. *Shiso* can be translated as "thought," "idea," and "philosophy," depending on the context or custom. For example "*shiso* shi" ("shi" means history) is translated as "history of ideas," and "postmodern *shiso*" as "postmodern thought."

Glossary

*1 Become cybernated: Cyberspace refers to a society or world constructed in a virtual computer space by creating on a computer things corresponding to real-world physical objects. When a physical object is copied (moved) into cyberspace, it becomes cybernated. As more objects become cybernated, cyberspace will become the real space, producing things nonexistent in the "physical" real world. In a broad sense, the monetary system and financial market are forms of cyberspace.

*2 Jackson method: A software design method proposed by M. Jackson, a prominent London-based software consultant. This technique is recognized as one of the first software design methods. In this method, a programmer first analyzes the data flow, and then uses the results to design software. Today, the Jackson method is rarely used because the programming environment has significantly changed.

*3 Component-based development method: A relatively new software design method. Like Dell Computer's strategy for computer production, which comprises procuring standardized parts and assembling them, this method entails creating standardized software components, which are then assembled into software.

*4 SAGE system: An early nuclear defense system of the U.S. It was the world's first system that delivered the common features of today's information technology, such as networking, real-time control, interactiveness, and graphic displays. Although the SAGE system served as the driving force of IBM's growth, it was hardly used in practice. Being designed for defense against the Soviet Union's nuclear bombers, the system became useless with the advent of the ICBM era.

*5 Hilbert Program : A mathematical research project carried out in the 1920s by Hilbert, the 20th century's greatest mathematician. The goal of the program was to provide secure foundations for all mathematics. Godel's discovery of the incompleteness theorem put an end to the program.

*6 Double major system : A system in American universities that allows students to concurrently major in two subjects. This is so common in the U.S. that American students studying in Japan sometimes feel unhappy about their being permitted to major in only one subject. A system similar to this is the major-minor system.

*7 Design Patterns : A collection of software construction patterns inspired by architectural design pattern collections compiled by the architect Alexander. The concept of Design Patterns is characterized by the use of pattern language, which acts as standard templates for description.

*8 UML (Unified Modeling Language): A specification language that is becoming a standard language for software design. Three representative methodologies were integrated to create this language, thus "Unified." With certification systems started in Japan and China, UML is almost a requisite for software engineers.

References

[1] Third Science and Technology Basic Plan 2006: http://www.mext.go.jp/a_menu/kagaku/kihon/main5_a4.htm

[2] Hayashi, Susumu, and Kurokawa, Toshiaki, "The Two Rationalities and Japan's Software Engineering," Science and Technology Foresight Center, National Institute of Science and Technology Policy, Science & Technology Trends- Quarterly Review, No.14, January 2005.

[3] Hayashi, Susumu, "New Trends in Software Development Methods," Kogaku (Japanese Journal of Optics), Vol. 34, No. 8, pp. 416-419, 2005.

[4] Cockburn, A., "Agile Software Development" (Japanese translation), Pearson Education Japan, 2002.

[5] Jackson, M., "Problem Frames: Analyzing and Structuring Software Development Problems," Addison-Wesley, 2000.

[6] D'Souza, D., UML 2004, Tutorial 5, Model-Centric Enterprise Architecture, October 12, 2004.

[7] Waldrop, M.M., "The Dream Machine: J.C.K. Licklider and the Revolution That Made Computing Personal," Penguin, 2002.

[8] Kita, Chigusa, "The History of Ideas of the Internet," Seidosha, 2003.(Japanese)

[9] Umeda, Mochio, "A Theory of Web Evolution: The True Revolution Has Yet to Begin," Chikuma Shobo, 2006. (Japanese)

[10] O'Reilly, T., "What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software": http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

[11] Friedman, T. (Iwan Fushimi, trans.), "The World is Flat: A Brief History of the Twenty-first Century," Vols. 1-2, Nihon Keizai Shimbun, 2006.

[12] Kobayashi, Michio, "The Natural Philosophy of Descartes," Iwanami Shoten, 1996. (Japanese)

[13] Godel, K. (Susumu Hayashi and Mariko Yasugi, trans. and com.), "Incompleteness Theorem," Iwanami Shoten, 2006.

[14] Boehm, B., "A Spiral Model of Software Development and Enhancement," ACM SIGSOFT Software Engineering Notes, Vol. 11, Issue 4, pp. 14-24, 1986.

[15] Kline, S.J., "Innovation is not a Linear

Process," Research Management, Vol.28, No.4, pp.36-45, 1985.
[16] "The Seventh Technology Foresight: Future Technology in Japan toward the Year 2030" (NISTEP Report No. 71, Jul. 2001), National Institute of Science and Technology Policy, Ministry of Education, Culture, Sports, Science and Technology, 2001.
[17] "The 8th Science and Technology Foresight Survey: Delphi Analysis" (NISTEP Report No. 97, May 2005), National Institute of Science and Technology Policy, Ministry of Education, Culture, Sports, Science and Technology, 2005.
[18] Chuma, Hiroyuki, "Complexity and Organizational Limits in the Japanese Microlithography Industry," Kogaku (Japanese Journal of Optics), Vol. 34, No. 8, pp. 388-395, 2005. (Japanese)
[19] Bälz's speech is partially available at the following Web page (Japanese):
http://med.main.teikyo-u.ac.jp/lab/kasahara/baelz4.htm
The original is "Bälz no nikki (Diary of Bälz)" (edited by Toku Bälz), Iwanami Bunko, which is out of print.

Susumu HAYASHI, PhD
Affiliated Fellow, NISTEP
Professor of Humanistic Informatics, Contemporary Culturology Department, Graduate School of Letters, Kyoto University,
susumu@shayashi.jp
http://www.shayashi.jp

The author studies informatics (information sociology, information science, information engineering) and science history. In particular, he has recently focused on sociological analysis of technologies in software engineering, and historical research into 19th century mathematics as the historical background of such technologies.

(Original Japanese version: published in October 2006)