The CTstage®*1) Development Story

Katsuyoshi Koide Masami Tobari

In the mid-1990s when the word “multimedia” started to become popular, talk of whether it was possible to develop devices that could handle multimedia emerged among engineers engaged in computer-related development work.

In the case of Oki, although we actually started such investigations with only a few engineers, the situation required that we start by having everyone embrace a common awareness of what “multimedia” indicated.

Also, around that time in the U.S., the service field called “computer telephony integration (CTI)” was already becoming established, and a group of products named “UnPBX,” for which the word “multimedia” was routinely used, began appearing on the world stage. Thinking that this would serve as a guide to what we also wanted, we began our development of “CTstage” by studying UnPBX.

Release of CTstage V1.0 (World’s First Office CTI Product) (1996 ~)

What we sought was a device based on a computer in which a general-purpose operating system is installed, which features telephone exchange functions and the ability to handle multimedia flexibly with software. “UnPBX” pointed to this kind of device.

So, what types of multimedia should CTstage handle? What types of specific functions should be included in the telephone exchange function?

First, we had to determine a concept for CTstage to clarify our objectives when working out the product specifications.

At that time, electronic mail was becoming popular, especially among large companies, and that wave of popularity was just beginning to spread to mid-sized and small companies. In general, however, the “telephone” and “fax” cultures were still strongly entrenched. Because of these conditions, we thought we would be successful if we could ride the “wave of electronic mail popularity,” integrate these various types of media, and make office operations more efficient.

The concept we decided on was “unified messaging.” In other words, CTstage would be a device that aimed to “revolutionize office communication,” was based on CTI, and was linked with the Internet, intranets, and groupware to achieve messaging solutions.

Although there were various media, such as “voice,” “fax,” “still images,” “ moving images,” and “electronic mail,” that we wanted CTstage to handle, we considered the opinions of the developers and decided to concentrate on voice, fax, and electronic mail--the three major media types at that time--and create a product that achieves unified messaging.

(1) Which hardware to use?

We began development of a product that would be based on an open platform and achieve differentiation through software.

“Which hardware to use?”

Of course, we also had the choice of developing everything ourselves, but starting in-house development from that point would have required a considerable amount of time for development. With this plan, we would have been unable to realize a timely product release and thus would have been late in riding the wave of the times.

The answer was simple. “If something good is already available out there, why not use it?”

This may have been a simple decision, but for hardware engineers who believe that design work means developing products in-house, the decision was a bitter pill to swallow.

This new idea of “using good products regardless of whether they were developed in-house or outside the company” is still true for CTstage today, and it is a fact that today’s CTstage is supported by this “commitment to no commitments.”

Finally, to select the platform, we investigated PC servers from both within and outside the company, and we decided to use our own “ifstation MB,” which was just about to be released. The reasons for making this selection were because the number of I/O slots allowed four full-size boards to be mounted in the slots and because the ifstation MB was equipped with the latest CPU at that time, namely, the 200 MHz Pentium® *2) PRO, and had superior cost performance.

The problem was in selecting the hardware for “voice” and “fax.” At that time, as far as CTI products were concerned, company “D,” company “N,” and company “B” had released I/O boards that had these functions installed and supported the ISA (Industry Standard Architecture) bus for Windows-NT. These boards were called “telephony boards” and were released mainly in the U.S.

Although all the boards of all three companies were equivalent in terms of function, each company employed its own proprietary architecture because the market had just gotten off the ground.

*1) CTstage is a registered trademark of Oki Electric Industry Co. Ltd.
*2) Pentium is a registered trademark of Intel Corporation.

The determining factor in selecting which company's product to employ was which architecture would become the mainstream in the future. After much discussion, the engineers finally decided to use the product of company "D," which had the largest share worldwide. Even here, there was mention of "commitment to no commitments," a premise that reflected our intent, as a key part of our development policy, to consider multiple vendors when proceeding with product development.

For our hardware engineers, who until then had worked on mainly proprietary development products, their gut feeling toward combining products purchased from outside manufactures was that it was extremely simple and unsatisfying. However, immediately after development was started, they realized that they needed to modify this way of thinking.

Starting from this combination of PC server and telephony board, which already have functions built into them, our challenge was "From what viewpoints can we build in added quality?"

The first thing we confronted was the physical layout problem. Telephony boards are connected in a daisy chain with a special telephony bus for sending and receiving voice and fax data between the boards. We therefore had to use special cables to interconnect all telephony boards inside the PC server. There was no PC server in the world that addressed the question of how to lay out such cables. Moreover, we could not change the external shape of the selected PC server. In the end, we were able to lay out the cables by creatively adjusting the cable lengths and cable folds so that the cables fitted into the limited space.

The next problem was the discovery that certain specifications of the hardware did not satisfy our company's internal quality standards.

The problems that stood out were the heat generated in the device and the inadequate resistance to power surges.

Some of the telephony boards for the ISA bus had a current consumption exceeding 3A for one board. If we used those boards, the internal device temperature during operation could rise above the company's internal standard value.

"Can we modify the PC server?"

We discussed the problem with the PC server department and began an investigation on the possibility of adding a fan. Since no such option was available in the PC server standard, we decided to prepare parts necessary for a fan addition designed especially for CTstage. At that point, the PC server became a specialized machine for CTstage.

Furthermore, it happened that the telephony board could not satisfy our company's internal standard for resistance to power surges. Although we also negotiated with the vendor, their engineers and ours had such different ideas regarding how the design should satisfy the standard that we decided to implement a unique solution for CTstage regarding this problem as well.

Although we originally thought there would be no in-house development for this project, it turned out that we were only able to release and ship the product (Fig. 1) because we ourselves developed a surge protector and figured out how to connect the surge protector to the telephony boards.

In the previous paragraphs, we introduced some of the improvements that were implemented during product development. However, from the onset, we became painfully aware of the difficulty in developing a product in this way. Even though this development project involved putting together purchased products, we had combine hardware devices that were based on different design concepts. We also had to work out our own internal differences regarding the design concepts.

Incidentally, by the time the device was released, CTstage had become a specialized device and the developers did not want it to look like a PC server. We therefore also developed a special rack, as shown in Fig. 1, that could hold items such as the main PC server unit, an uninterruptible power supply (UPS), and the surge protector. Although this rack design later became a matter of debate, it was one of the contributions of the hardware engineers.

Fig. 1 CTI Intranet Series (Minitower type)

(2) Which software to use?

From the fact that CTstage requires telephony boards, it is probably clear that this product is dependent on hardware. CTstage, however, is also a computer-based communication system that was constructed from open platform products, so it was a product for which software was extremely important. Therefore, when the "go" was given for this product, we promptly began to prepare a software development organization. Since we assumed that the platform for this product would be a PC server, we had to find employees who had sufficient knowledge of Windows. Also, given the functional requirements, the team also had to know about mail systems and communications. Previously in such circumstances, it was common practice for engineers to get on with development while studying to catch up on the technical know-how they lacked. However, since this project required fast development, we decided to build an organization by conducting a wide-area search and selecting members who already had necessary technology knowledge. As a result, we focused on members who were already working on the Windows operating system projects and gathered members with technical skills in areas such as databases, mail systems, development environments, and communications. On April l, 1996, the “CTI Systems Group” was born.

When the group first got started, the group leader presented an overview of the project and announced the schedule. That schedule was an unprecedented one that called for shipping a product in about six months, and all members were dubious about the schedule. However, the group leader came out with the slogan, “Let’s build the best software in the world!”, and spurred the member’s enthusiasm.

Since the following factors were involved, instead of opting for the previously used “waterfall” model for software development, we decided to adopt the spiral approach, in which we would operate the prototype sequentially while fixing any problems that came up.

- The time to delivery was short.
- The software would be developed on an open platform, and there was the possibility of unexpected problems occurring.
- Since this was our software to be developed ourselves, it would take a while for the specifications to be finalized.

Since a minimum of two cycles is required for spiral-type development, the schedule called for (1 evaluation in three months. Through a continuous series of intensive review meetings, the functions and architecture of the software gradually came together.

Our review of the functional specifications was driven by the following two concepts:

- To revolutionize office communication through unified messaging technology.
- To develop an integrated computer/telephony system.

At first, we began by asking, “What is unified messaging?,” but as the investigation progressed, the role of unified messaging became clear. In this project, we defined unified messaging as a technology that centrally manages various media information and retrieves desired information through various means, and we investigated application functions that would handle the various media, such as fax (image), telephone (voice), mail (text), and web information, as they are being used in offices (Fig. 2). To implement applications, we employed Oki Electric’s media conversion technologies, such as text/voice synthesis, voice recognition, and translation. Fax-optical character recognition (OCR) technology was also a candidate, but was shelved because the technology still had many restrictions if we presumed that the product would be used in the office.

The investigation of the software architecture was driven by the following two concepts:

- “Simple is best!”
- The architecture used must last 5 years (use the latest technology)

The architecture made full use of the latest Windows technology. Although this may be obvious, we used the client-server model and adopted a method that allowed the client to invoke server functions through remote procedure calls. Also, we made the server functions visible in client applications as objects by using ActiveX controls. This was the original format of the CT-application program interface (CT-API) that would later be the key to CTstage superiority (Fig. 3). We revised the API several times before actually releasing it, but using this API has made it possible for custom applications to be developed at the user level. While it is not true that until now there were no CTI products that had an API function, even when we think about it now we believe that our CTstage is the only product that can provide the wide range of functions necessary for application building. These functions include not only the line system interface represented by TAPI, but also unified messaging, contents management, and media conversion.

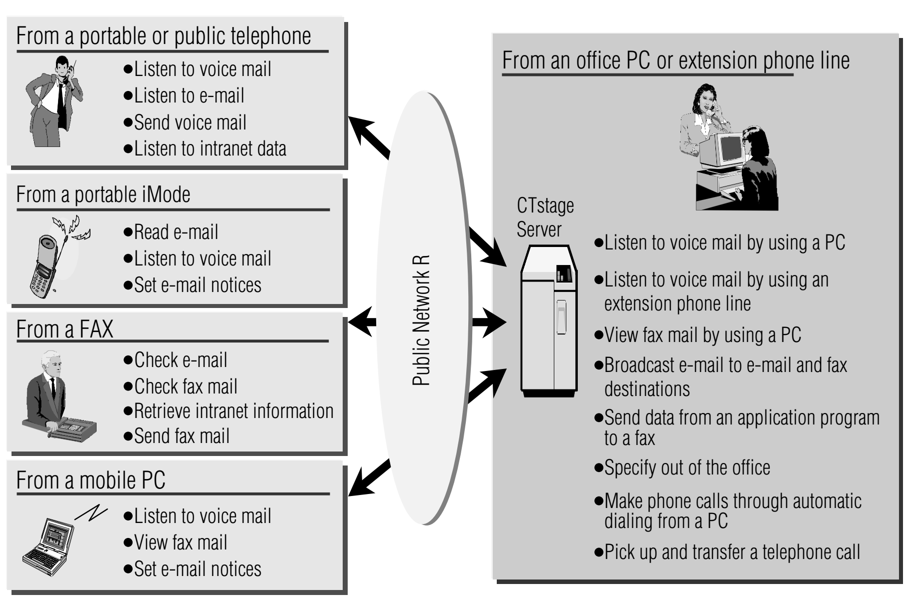

Fig. 2 Conceptual diagram of the unified messaging function

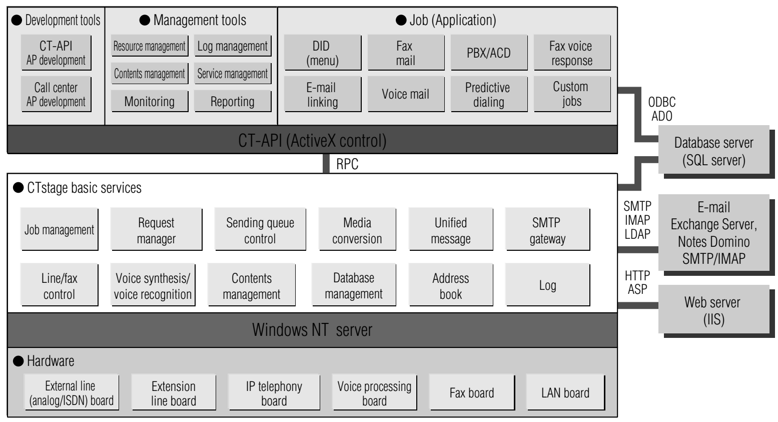

Fig. 3 CTstage software configuration

(3) The debut of CTstage

In September, 1996, the product announcement day drew nearer and nearer. Although development was running late, the unified messaging function had begun to operate. However, since the function was still not stable, scenarios were carefully rehearsed for the live demo on the day of the announcement. "How do we make a demo that will have both high appeal and a low chance of failure?" In particular, the voice commands that used voice recognition had some quirky recognition results and everyone was worried. Since there was the possibility of a large amount of ambient noise at the announcement site, the outlook was quite gloomy after watching the rehearsal conditions. We then came to the realization that this was only a demo and worked on tuning the recognition dictionary. In other words, we re-created the dictionary by optimizing it for the voice of the group leader, who would be doing the demo on the actual day. The results turned out to be fairly satisfactory, and the outlook for the product announcement demo became brighter. All this work was not done in vain, for on the day of the announcement, everything went unbelievably well, and CTstage got off to a favorable start.

About three months after the product announcement the voice recognition function had stabilized, and shipment of the product as the CTI Intranet Series began. Initially, however, the product was only available as a preinstalled model. Shipment as a software package began with V1.1, along with the release of the CT-API mentioned above. This V1.1 product became the base for the current CTstage.

Release of CTstage V2.0 (Integration of Office CTI and Call Center) (1998 ~)

CTstage V1.0 was initially released as a minitower unit that could accommodate up to 8 external and 16 extension analog lines, and 4 fax boards. After V1.0 was released, we received many requests to diversify the line accommodation scope and to offer a greater variety line types. Thus for the next development stage, we decided to expand the line-up to support the whole range, from small-scale to large-scale line configurations.

For the number of lines to be accommodated in this new development, we decided on a line-up that expanded the total number of external/extension lines from the maximum of 24 lines (8 external lines and 16 extension line) of V1.0 to a range starting from 4 lines (4 external lines) and extending to 98 lines (48 external lines and 48 extension lines). We also began looking for platforms on which to implement this line-up.

To implement this specification, we needed a model that could hold a single telephony board and a platform that could hold up to 10 boards.

First, for the platform of the small-scale model, we decided to use a desktop personal computer to keep the cost low. This was a model for bringing unified messaging to small offices, specifically to handle the proliferation of voice mail, and we also wanted it to be compact.

However, as the platform investigation progressed, we discovered that there were surprisingly few desktop personal computers that could house a telephony board since all telephony boards are full-size ISA boards. The main practice at that time in regard to optional boards for personal computers was to use half-size boards, most notably the LAN board, so PC vendors did not need to offer a design that would allow installation of full-size boards.

Although we had hoped for a compact model, in the end we had to settle for a relatively large desktop personal computer, as shown in Fig. 4.

An even more troublesome task was selecting a platform that could house up to 10 telephony boards.

Fig. 4 Desktop type

We looked for platforms both within and outside our company, searching the Internet, magazines, conventions, and also holding meetings with PC server vendors. For the potential candidate platforms, we actually tried real units optional boards for personal computers and conducted a physical verification of whether the telephony boards could be installed.

One of the problems was that, even if the catalog indicated that a platform supported full-size ISA boards, when we actually tried to install the telephony boards, the boards could not be installed because of slight interference with other parts. In the end we found that there was no commercially sold PC server that could satisfy the board installation requirement. What we then turned to, as candidates, were industrial-use PC servers.

Industrial-use PC servers have the following advantages: they were designed on the assumption that parts would be customized; compared with commercial servers, they have more flexibility in the number of boards that can be installed; and because they use embedded CPUs, the period during which they are supplied is long, compared to commercial servers.

However, these advantages also came with a disadvantage. Compared to commercial PC servers, industrial-use PC servers lagged about a year behind in adopting the latest CPU trends.

Actually, the first CPU that we used was the Pentium® II 333 MHz CPU, and subsequently we used faster CPUs as appropriate. However, the performance improvement in the CPUs used by commercial servers was accelerating, and after one year, there was a big gap in the CPU specifications. Of course, device performance is not determined solely by the CPU performance. However we worried that in some cases there might be a perceived performance inadequacy, just based on the catalog specification values.

To address this concern, each time the CTstage software version was upgraded, the Engineering Department ran actual device verification tests, exercising the software as well, and showed that industrial-use servers had sufficient performance to be operated as CTstage. This proved that the commercial-use servers were devices that we could have customers use with confidence. We had known this when we selected an industrial-use server for CTstage. However, as it turned out, we painfully realized that the market saw CTstage as a type of commercial server, even though it was actually a specialized machine equipped with telephony boards and special software.

Fig. 5 shows the rack mounted-type based on use of an industrial-use PC server.

There were various software changes in V2.0, but the main ones were the larger number of circuits to match the new hardware configuration and support for a call center function. Whereas we previously offered a capacity of 24 analog circuits, we now used INS1500 and INS64 for a maximum of 92 circuits. Of course, since the machine specification had been raised since V1.0, it was also necessary to improve the software performance. In particular, we improved the circuit processing portion to use a table access method, so that circuit control would not be serialized.

Fig. 5 Rack mounted type

When we started developing V1.0 there were already many manufacturers of call centers in the market, and know-how was required. As a result, we decided it would be difficult to incorporate that function into a latecomer product like CTstage, and we stayed away from that market. After V1.0 shipped, CTstage acquired a good reputation both in Japan and overseas, but sales still did not meet the target. It was impossible to do something about this while ignoring the call center market, so we quickly changed our plans and moved our product toward the call center market as well. This capability for rapid change is one of the real advantages of CTstage.

Even though we called it a “call center,” CTstage was in fact a latecomer, as explained above, and rather than competing head-on we made use of the advantages of the inexpensive platform UnPBX and set our sites on providing a small, informal call center.

It seems obvious today, but the system used features such as a number display service so that the operator could see customer information as soon as the call came in (Fig. 6, Fig. 7). The key to a call center is ACD (automatic call distribution). Since we lacked know-how we developed ACD by studying similar systems. There were American products that were quite good for the ACD basics, and we made it a goal to have our specifications surpass such products.

We incorporated the ACD function by expanding the PBX job that had been developed for the unified messaging function. Also, since ACD needs to link with applications running on the operator’s terminal, we gave CTstage an API similar to the CT-API.

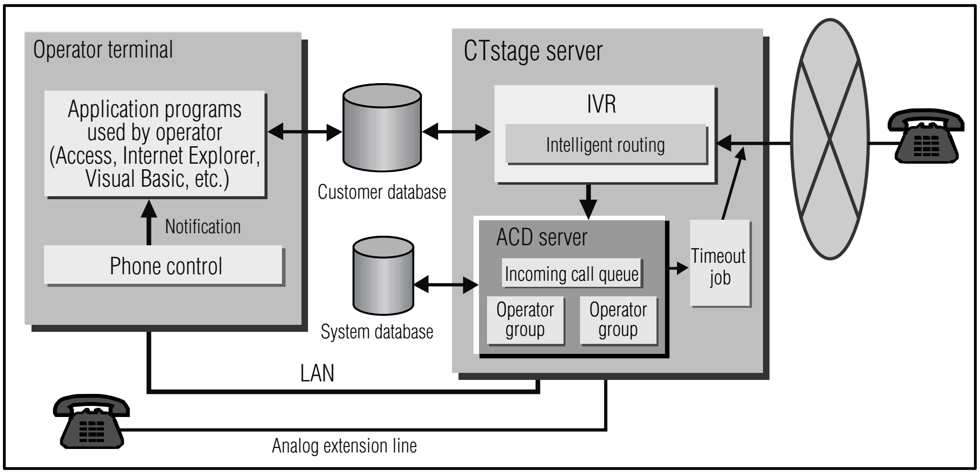

Fig. 6 Conceptual diagram of the call center system

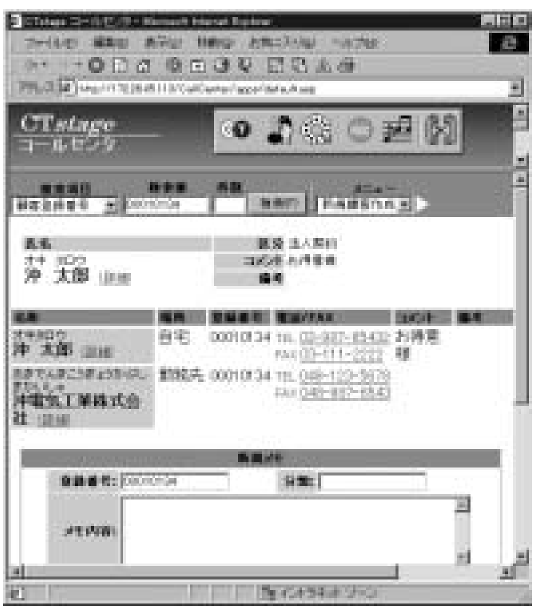

Fig. 7 Operator screen displayed by Internet Explorer

Even with just these functions, CTstage would have been a very innovative product for that time. In fact, however, CTstage ended up being an even more advanced product because we also made the client application a web application.

Release of CTstage V3.0 (Distributed Call Center) (1999 ~)

While CTstage V2.0 increased the number of extensions to 48 lines, there was a demand for even more lines, and CTstage V3.0 responded to that.

In V3.0, we added a function that enabled telephone calls and data to be transferred between multiple CTstage machines, and developed a new architecture that made multiple CTstage machines to appear as one machine.

To realize these functions, we developed a method that transfers data by using the ISDN data transfer function called ISDN User-User Information (UUI). We also developed a method that transfers telephone calls and data by connecting CTstage units through a LAN and using IP telephony.

By applying these methods to the call center system, we could realize real-time transfer of telephone calls and client data between multiple CTstage machines. These methods therefore provided the means for constructing large systems from smaller building blocks and for creating a virtual call center in which call center devices installed at remote locations appear as a single system.

Around 1999, Voice over Internet Protocol (VoIP) was being mentioned as an important new technology. Since we had a passion for new technology when it came to CTstage, we began studying VoIP right away. At that time, the main purpose for applying the VoIP application was to reduce telephone fees. However, for an application server like CTstage there is no particular advantage in economy alone, so we examined VoIP from the viewpoint of whether it could be used in improving our services.

Meanwhile, the large number of inquires we received regarding the call center function added in V2.0 prompted us to start a plan for expanding the call center function. This plan would allow us enhance the call center function from one geared toward small-scale call centers to one capable of handling more than 100 operator desks. Since CTstage was based on a PC platform, there was a hardware limitation on the number of lines that one PC could accommodate. We therefore came up with the idea of building a larger system by linking distributed servers. For the linking method, we thought of using the ISDN data transfer function called ISDN UUI to transfer customer data. However, using ISDN meant that we could expect operational difficulties since we would have to use a public network or ISDN extension lines in a PBX. That is when we thought of using VoIP as the transfer method (Fig. 8).

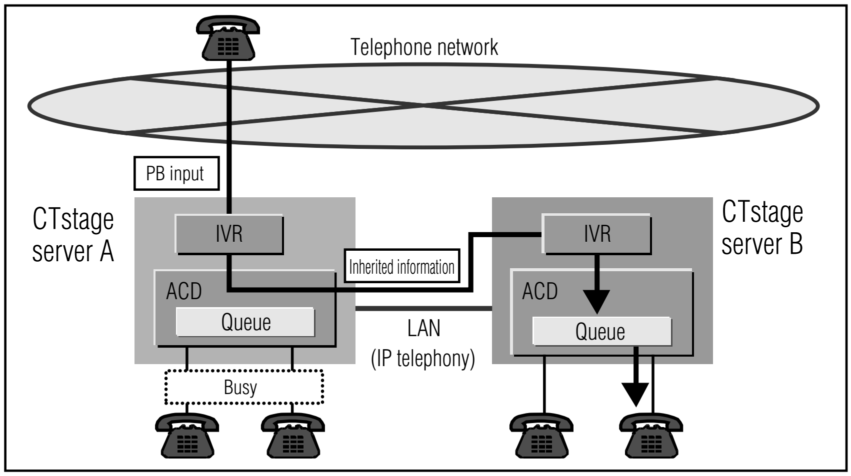

Fig. 8 Conceptual diagram of a distributed call center

Using VoIP meant easy installation because a PBX and other equipment would be unnecessary. Although there was a problem with bandwidth, the merits of using VoIP were large for a distributed call center because it allowed location flexibility.

Thus we began to consider using VoIP mainly for the call center function. Although VoIP made linking between servers possible, it would also place a constraint on the ACD function because operator management would be carried out at each server. After investigating methods of increasing the number of operators that one server can handle, we thought of using VoIP instead of analog extension lines to connect the operators and the servers. Then, even if the number of operators increased, the need for sharing terminals would be eliminated, and operator desks could be set up in remote locations by linking the desks through an IP network. These decisions consolidated our plan, and we quickly went to work on implementation.

Although the VoIP technology presented in V3.0 was often used for distributed call centers in actual operations, the technology was still not used much for VoIP-connection of operation extension lines. This area would be completed in CTstage4i.

Toward CTstage4i for .NET (the challenge to achieve even larger scale systems) (2002 ~)

The next step for CTstage is CTstage4i for .NET, which will feature a new architecture.

To handle the needs of an era headed toward broadband and IP processing, we have substantially strengthened CTstage's affinity with the Internet.

There are two CTstage models, an UnPBX model and a softswitch model, and the new CTstage version is designed to handle the call center system functions of large scale installations of over 300 operator desks.

The PC server we selected uses a leading-edge architecture that was developed by IBM for CTI market servers and houses up to 12 telephony boards (Fig. 9). Furthermore, in line with the concept of "using good products regardless of whether they were developed in-house or outside the company," we formed alliances with various vendors, including peripheral device vendors, and built an organization that will enable us to enrich the functions provided by the system.

Fig. 9 CTstage4i server

For the UnPBX model, which is already being shipped, the basic architecture has not changed, but Windows.NET technology was introduced, and a new system database and new system management tools were developed. We had anticipated certain risks to accompany use of a new technology. In addition, however, we encountered problems that affected sections for which only the scope was increased while the software was not changed at all. For example, CTstage normally starts an application called a job in each line, but sometimes this job could not be started because of a Windows resource restriction. We submitted an inquiry to Microsoft regarding this condition and found out that there are tuning parameters that can be adjusted. However, since there was no procedure for determining the optimum parameter values, we ended up identifying the optimum values by trial and error. Here again, we were painfully reminded of the difficulty in building a system on an open platform. The softswitch model, which is a major attraction of CTstage4i, has just entered its final stages of development. Shipment of CTstage4i has already started, and we have high expectations for the system because CTstage4i shows great promise in many aspects.

Toward Further Growth

Fig. 10 shows the transitions in the development of CTstage. CTstage, a device that must constantly evolve, is the result of cumulative efforts. As proof of the effectiveness of this approach, CTstage was selected as one of the “HOT 5,” a designation awarded to superior products and services at the Computer Telephony World Expo, a CTI exhibition sponsored by IDG Japan. CTstage went on to receive “THE BEST OF CT World” award, which is bestowed upon the best product in the computer telecommunication field, for three years running, and it is currently esteemed as the top brand in the CTI industry. (Fig. 11)

On top of this success, there are even things we want accomplish in the future. These include branching out from enterprise services into public services (large-scale systems) and expanding to SOHO services (small-scale systems), which previously could not be realized with the release of our desktop-type unit.

Please look forward to watching how CTstage will evolve in the future.

Authors

Katsuyoshi Koide: Multimedia Messaging Company, Customer Support Div., General Manager

Masami Tobari: Multimedia Messaging Company, CTstage Development Div., General Manager

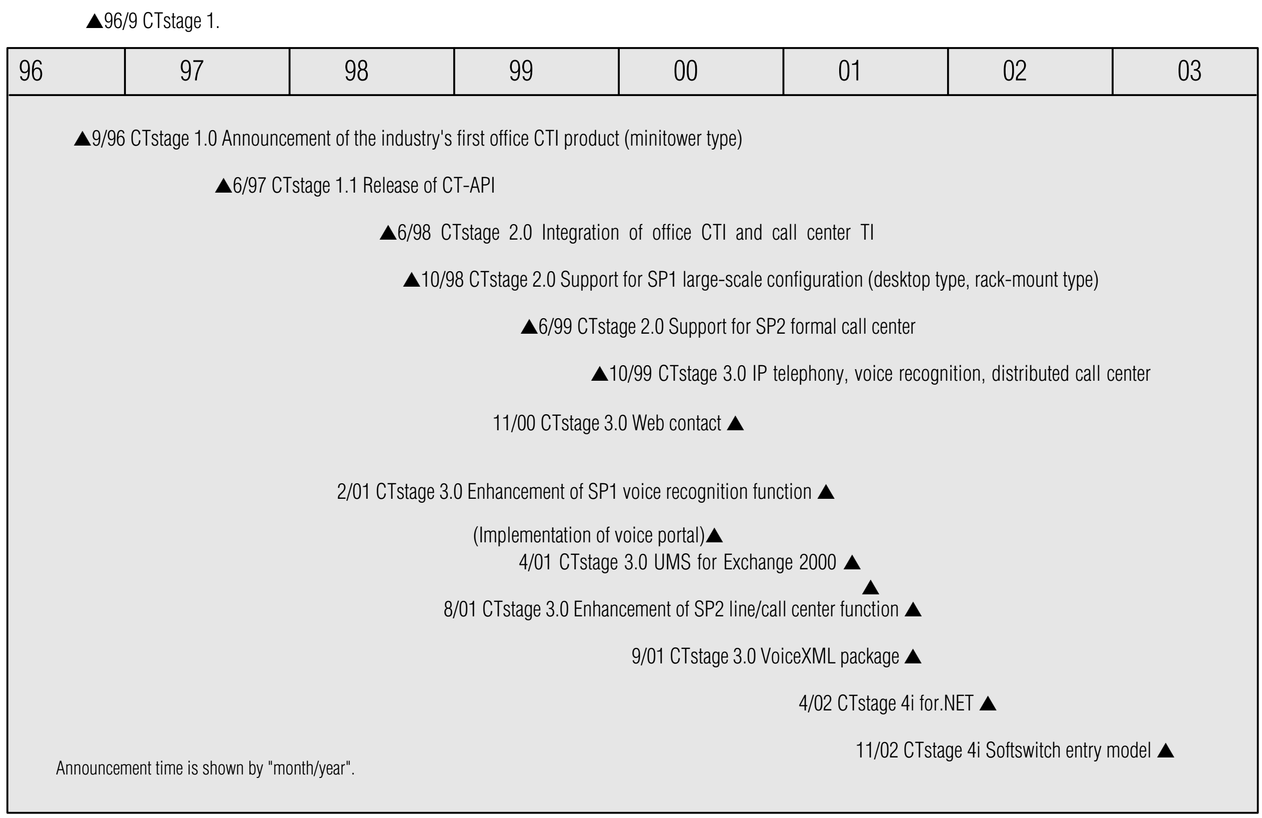

Fig. 10 Transitions in CTstage development

Fig. 11 “THE BEST OF CT World” awards