The alternate support element, a high-availability service console for the IBM eServer z900

by B. D. Valentine
H. Weber
J. D. Eggleston

The alternate support element feature provides a redundant service console attachment for the IBM eServer z900 that improves the availability and serviceability characteristics of the z900 for manual operations and service/maintenance tasks. All of the tasks relying on the zSeries™ support element implementation (such as activation, configuration, and repair functions) are gaining inherently high availability characteristics with the provision of an alternate support element. The functional concept is physically based on two standard IBM ThinkPad® computers which are packaged with a z900 server and act as service consoles and system controllers for a zSeries system. The logical relation between the two consoles is based on a master/slave concept, together with a failover design that provides the capability for manual and automatic (i.e., under program control) switchover from the primary support element to the alternate support element. Also included in the concept are automatic role determination for primary and alternate roles, a synchronization mechanism between the two support elements based on mirroring, and a design which will allow the alternate support element in the future to act as a staging area for updates and pre-loads of Licensed Internal Code (microcode).

Introduction

The alternate support element (SE) feature was initially released on the IBM 9672 G6 series of processors (S/390 Parallel Enterprise Server* Generation 6 processors). The IBM 9672 family of servers (hereafter referred to as 9672) previously had only one SE per system to provide such functions as system activation and loading of an operating system (i.e., OS/390*, VM, etc.). If the functioning of this SE was disrupted, the operating system(s) running on the 9672 would continue, but a new partition could not be activated, nor could an operating system be reloaded until the SE was repaired (which could take several hours if an on-site visit of a service representative with the repair parts was required).

When the 9672 family of servers was beginning to replace high-end bipolar systems with CMOS systems, a

0018-8646/02/$5.00 © 2002 IBM

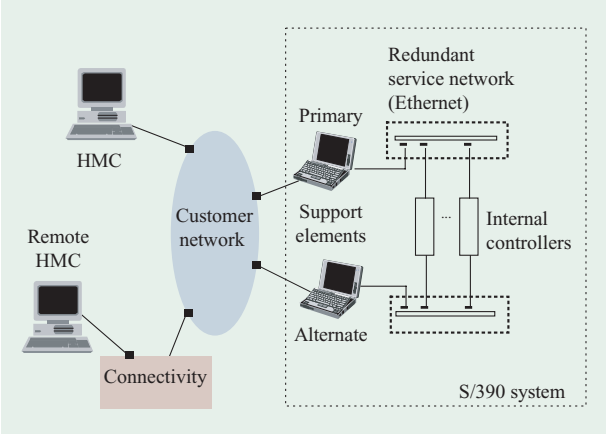

Figure 1

z900 alternate support element.

"redundant SE" implementation (as in the bipolar systems) [1] was requested, and it became a requirement to provide a solution for this "deficiency" of the 9672.

A second redundant SE (called the alternate SE) is now provided in addition to the primary SE (the active SE) to provide continuous SE availability. When the primary SE fails, the alternate SE is switched to make it the new primary SE, and the failed SE becomes the new alternate. This allows the repair action to be performed on the alternate SE concurrently with customer operations at a later time.

Periodically, the primary SE mirrors its changed file data to the alternate SE to keep the alternate SE hard disk at the same running state as that of the primary SE. The data is mirrored across the service network [the Ethernet local area network (LAN) connecting the SEs and the z900 processor] or the customer network [the LAN used by the hardware management console(s) (HMCs) [2] and the SE for communication].

It is important to note that the primary SE is the only SE used for communication with the 9672 processor. The alternate SE has no zSeries* functional responsibilities other than recovery [3] for situations in which the primary SE becomes unavailable for customer operational tasks.

Switching

9672 S/390 G6 processor

For the S/390* G6 processor, both the primary and alternate SEs had cables attached to their parallel port adapter on the SE ThinkPad* hardware, and these cables were routed to a physical switch which then had a single cable output connected to the 9672 processor. If the primary SE failed, it would trigger an error that was reported to the service provider. If the customer's data center was at a distant location, the service provider would have to send a support person to the data center to throw the alternate SE switch and force the alternate SE to become the primary SE and the former primary SE to become the new alternate SE. If the data center was situated locally, the service provider could also have asked the customer to go to the 9672 to throw the switch as well.

Once the alternate SE switch was thrown, the SEs reversed their roles, regardless of their current states. If the primary SE was busy (possibly with service actions) or the alternate SE was at a microcode level that was possibly incompatible with 9672 processor microcode, the switch would still have executed. There was no way to stop or to control the switch, since it was a physical entity with no supporting software logic.

Also, in many cases the 9672 systems are locked and unattended in secure rooms, and it can take time to get a service representative to the location of that system and obtain access to it. This is unfortunate, since the HMC allows the customer to operate the 9672 remotely, not only from another location in a building but also from almost any location around the world.

z900 processor

For the z900 processors, IBM has provided solutions for each of the limitations of the alternate SE implementation, as discussed in the previous section. **Figure 1** shows an overview of the z900 alternate SE solution, including locally and remotely connected HMCs. The following is a high-level description of the z900 alternate SE, but it could be applied to any redundant console attachment requiring high availability. The next section provides more detail on the intelligent switchover controls used to attain high availability on zSeries, and **Figure 2** shows how this could be applicable to any redundant console attachment.

For the z900 system, the SE no longer uses the parallel port adapter as the connection from the SE to the z900. Instead, an Ethernet LAN is used as the connection, allowing both the primary and alternate SEs to always be physically attached to the z900 processor. Only a single SE, the primary SE, is still used as the active SE for performing customer operations such as activation.

There is no longer a physical switch to initiate switching between the primary and alternate SEs. The z900 now provides a software switch for controlling which SE is the primary SE, and controls for managing this can be operated from the HMC (local or remote), the primary SE, and the alternate SE. In addition, an automatic switchover (failover) can occur if certain error conditions are detected. Both the user interface and the automatic switch can be disabled based on the state of the SEs. In addition,

a user interface switch may be concurrent or disruptive to operations of the z900 processor based on the microcode levels on the alternate SE hard disk.

Intelligent switchover

The z900 alternate SE switch design is innovative in the robustness of intelligent switchover for the following areas:

- User interface switch with state checking.
- User interface switch with concurrency controls.
- Automatic switch with state checking.
- Remote user interface switch with state checking.

Figure 2 illustrates how the z900 alternate SE intelligent switchover solution can be extended, with high availability, to any redundant console attachment. The remaining sections provide the specific details of the z900 solution.

User interface switch with state checking

The user interface switch is a manually initiated customer operation or a service action, and in the past, the switch would always be allowed to execute. Now, manual switching is not allowed to occur if any of the following conditions (manual switch constraints) exist:

- The last changed primary SE state information was not successfully received at the alternate SE.
- The SE is inoperable (fenced) as a result of a previous automatic switchover.
- The engineering changes task is in progress.
- The hard disk restore task is in progress.
- The application of Licensed Internal Code changes by the change internal code task is in progress.
- Mirroring to the alternate SE is in progress.
- The primary SE cannot communicate with the alternate SE. (However, if the alternate SE cannot communicate with the primary SE, switching from the HMC or alternate SE is allowed.)

The user interface switch panel function can be invoked from any of the user interfaces of the HMC, primary SE, or alternate SE. The corresponding user interface state checking is performed at the primary SE or the alternate SE, depending on which SE handles the functional switch request. The location of the functional request handler may be the primary SE or the alternate SE depending on whether the primary SE is operational and which user interface has been used to invoke the panel.

This switch state checking prevents the user from switching to the alternate SE while certain operations are in progress that may jeopardize the data integrity of the SEs and/or the z900 processor. It also does not allow the user to switch to a SE that may be in a worse state than the current primary SE.

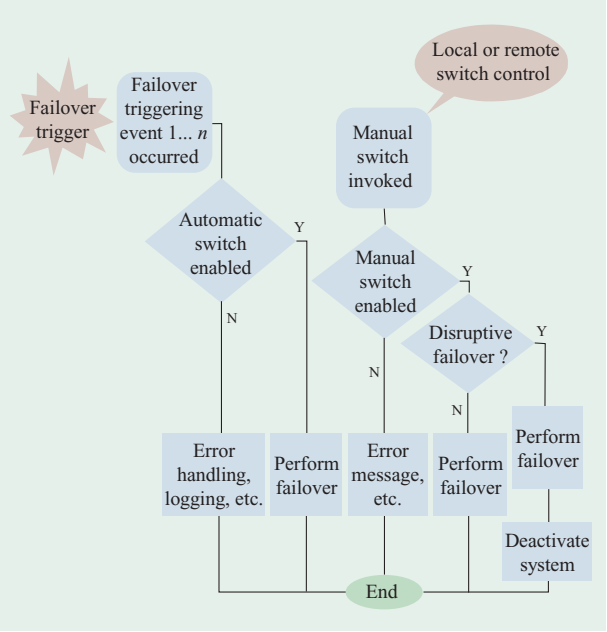

Figure 2

Control flow for support element switch mechanism.

User interface switch with concurrency controls

In addition to the user interface switch state checking, there is also an intelligent concurrency control to ensure that if the alternate SE is switched to the primary SE, it can continue to interface correctly with the z900 processor microcode. Since the primary SE hard disk contains all of the microcode that runs in the z900 processor, there may be prerequisite and/or corequisite interfaces between the z900 processor and the SE microcode. When these changes are applied to the SE either via change internal code or engineering changes, the data will eventually be mirrored to the alternate SE hard disk via alternate SE mirroring or hard disk restore.

However, there may be an extended period of time (up to 12 hours) for which the microcode data on the primary SE and alternate SE hard disks could be different. During this time, if a switch were to occur, the new primary SE microcode might not support the new interfaces to the z900 microcode, and SE communication to the z900 microcode might simply not work or, in the worst case, could even cause data integrity problems in the z900 processor.

To solve this problem, alternate SE switch concurrency controls were added to provide concurrent switching in addition to disruptive switching. If the microcode data between the primary and alternate SE hard disks matches, the alternate SE switch operations can be performed concurrently with z900 processor operations. If there

is a mismatch, the alternate SE switching can only be disruptive to the z900 operations. In the disruptive case, the z900 processor is powered off, basically to unload the current z900 processor microcode, and the z900 processor microcode is eventually reloaded to match the new primary SE hard disk corequisite/prerequisite interface requirements.

Automatic switch with state checking

The automatic switch control offers two new aspects: the automatic switch itself and the state checking to allow it to switch. The automatic switch is triggered by any of the following events:

- The primary SE detects that it has a hardware problem internal to itself (i.e., a diskette drive problem or some other minor hardware problem that allows the primary SE to continue to run, but it should switch to the alternate SE until its internal problem is fixed).
- The primary SE cannot communicate with the z900 processor.
- The alternate SE detects that the primary SE is no longer operational.

However, the above automatic switch trigger conditions will not cause an alternate SE automatic switch to occur under any of the conditions listed in the section on the user interface switch with state checking. This implies that the "automatic switch enabled" checks shown in Figure 2 include the "manual switch enabled" validations as a subset. In addition, the alternate SE automatic switch will not occur if any of the following conditions exist:

- The microcode data on the alternate SE hard disk is different from that on the primary SE.
- The alternate SE cannot communicate with the z900 processor.
- A user is logged on to the primary SE in product engineering (PE) or service mode.
- The primary SE has service status enabled.
- The user has configured the automatic switch to be disabled.

Again, this condition checking prevents the automatic switch from occurring while certain operations are in progress that could cause the data integrity of the SEs and/or z900 processor to be in jeopardy. It also does not allow an automatic switch to a SE which may be in a worse state than the current primary SE.

Remote user interface switch with state checking

z900 servers also provided a way for the user to invoke an alternate switch from a HMC (local or remote) whether or not the primary SE is operational. The HMC normally has a communication link to only the primary SE and has no communication with the alternate SE. If the HMC can "talk" to the primary SE, the user interface switch works as if it were invoked locally at the primary SE. This means that the state checking conditions listed in the section on the user interface switch with state checking are validated before switching is allowed to occur.

If the primary SE is not operational, the HMC creates a temporary communication link to the alternate SE, and the HMC-initiated switch request is then sent to the alternate SE. Again, the state checking conditions for the user interface switch are confirmed before switching is executed.

In both of the HMC cases cited above, the HMC could be either local to the z900 system (in the same location/local LAN as the SEs) or remote to the z900 system (different location/same or different LAN from the SEs). This is important, since this allows switching to be performed without going on-site. In many cases, the z900 processors are operated remotely, and it was important to provide remote alternate SE switch capability including the state checking.

Mirroring

An important requirement to be prepared for an alternate SE switch is to ensure that the primary and alternate SE hard disks are kept synchronized. This is accomplished via an alternate SE mirroring operation via a scheduled mirroring request, an immediate mirroring request, or an automatic mirroring event.

Scheduled mirroring

Each day at 10 AM and at 10 PM, scheduled mirroring events occur in which data from the primary SE hard disk are automatically mirrored to the alternate SE. In general, if primary SE changes [such as Licensed Internal Code changes (patch) or profile customization changes] are initiated from the HMC and SE, the immediate mirroring function should be used to ensure that those changes are not lost in the event that an alternate SE switch may be required (as a disruptive switch in the case of a primary SE failure) prior to the next scheduled mirroring event.

When primary SE changes are initiated from other sources such as hardware configuration definition (HCD) writing updated I/O configuration data sets (IOCDSs) to the primary SE, it would also be appropriate for the operator to initiate an immediate mirroring. In general, the scheduled mirroring function is a safety net for changed data to be synchronized to the alternate SE.

Immediate mirroring

The HMC and primary SE provide a user panel to initiate an immediate mirroring of data from the primary SE to

the alternate SE. As already mentioned, when certain HMC and SE tasks are performed (Licensed Internal Code changes, customizing profiles, cryptographic coprocessor configurations, or any task that causes changes to the primary SE hard disk), it is recommended that an immediate mirroring be initiated. This should be convenient for the user, since he or she is already at the HMC or primary SE console.

However, if it is already close to the scheduled mirroring times of 10 AM or 10 PM, the user may simply choose to wait for the scheduled mirroring event rather than take the time to do the immediate mirroring.

Automatic mirroring

Only one function currently triggers an automatic mirroring—the rebuild of the primary SE vital product data (VPD). This VPD rebuild can be initiated from a HMC or SE user panel, or it can execute automatically when the primary SE detects that the system hardware has changed. Since the rebuild VPD function may execute several times during the process of changing hardware, it triggers the alternate SE mirroring function automatically one hour after the last rebuild VPD execution.

Automatic mirroring was provided for the rebuild VPD function, since it is extremely important to keep the primary and alternate SE hardware configuration views synchronized; otherwise, an alternate SE switch may not be successful. However, automatic mirroring triggers were limited to this rebuild VPD function for a few reasons. First, when mirroring is executing (which typically requires about 15 minutes), alternate SE switching is blocked; this would have an impact on an alternate SE automatic switchover when a problem is detected. Second, if the automatic mirroring were extended to several functions, there could be too many requests for mirroring. Finally, the two mirroring events that are already scheduled each day seem to be sufficient to get less-critical data mirrored if no one has initiated an immediate mirroring.

Mirroring protection

Certain conditions require the primary SE to prohibit the execution of mirroring in order to protect against possible data corruption:

- The system may be in the middle of changing data.
- A service representative may be in the process of repairing the system.
- An automatic alternate SE switchover could have occurred, with error data remaining on the fenced alternate SE (previously the primary SE).
- The primary and alternate SEs may be at different Licensed Internal Code engineering change levels.

For the last case, the alternate SE provides a quick recovery path to return to the previous engineering change level. Mirroring is unblocked for this condition once the alternate SE is reloaded to the same engineering change level. The alternate SE is loaded by first doing a backup critical data function of the primary SE from a HMC to a DVD-RAM; then a hard disk restore from the HMC uses the DVD–RAM primary SE data to copy an exact image of the primary SE hard disk to the alternate SE hard disk. Mirroring does not copy an exact image of the primary SE hard disk to the alternate. More details are given in the next section.

Mirroring improvements for the z900

The G6 processor mirroring uses the customer network to transmit the mirroring data. Generally, the amount of data sent is not too large, but there is sometimes a substantial amount of data sent, and this could cause some performance degradation to the customer network. For z900 systems and the addition of the service network (Ethernet LAN connecting the SEs and the z900 processor), the data is now mirrored across the service network. As a recovery path, the data will be mirrored across the customer network if communication across the service network is not working, but this should be rare.

Communication between primary and alternate SEs

Primary/alternate SE selection

The support element provides multiple paths, all leading to its own initialization: the power-on sequence of the ThinkPad, a reboot of the ThinkPad (which may also be initiated remotely from a HMC), a local keystroke combination, and an internal recovery function. During this initialization, the SE must determine whether it is to be a primary or an alternate SE. It is very important to the customer to ensure that whichever SE was last primary remains as the primary SE, since there is a possibility that the two SEs contain slightly different data (if they had not yet been mirrored after data was customized on the primary SE, or if an engineering changes task or a Licensed Internal Code changes task had been performed placing one SE at a different microcode level than another SE). An exception to this would be if the primary SE were dead (hardware problem).

During the time during which a SE is trying to determine its role as primary or alternate SE, the SE is in a state called *preSE*. A file called the *soft switch* file on the SE hard disk is used to identify which SE was last primary. While in the preSE state, the SE first tries to communicate across the service network using TCP/IP to determine whether the other SE is already primary or alternate SE or is in the preSE state, and whether or not

it has the soft switch file. If the other SE already has a defined role as primary or alternate, the preSE takes on the opposite role. If the other SE is also in a preSE state, it uses an algorithm based on the existence of the soft switch file on each of the preSEs to determine which SE should become primary and which should become alternate.

Network path

Generally, the network communication path used for communication between the primary and alternate SEs is the service network. However, if for some reason this network path has a problem which does not permit positive communication between the SEs, the SEs will use a recovery path via the customer network. Since it is not desirable to place additional network traffic on the customer network, the primary SE will try every minute to revert back to using the service network for primary-to-alternate-SE communications. If this recovery to the service network is not successful, the communications will remain on the customer network. When the service network communication path fails, an automatic problem report is made which should trigger a service representative to investigate the problem at the machine, and using the customer network as a backup communications path ensures that the functionality of the alternate SE remains intact.

Loss of communication between SEs

As discussed above, if there is a loss of communication between the primary and alternate SEs, the customer network is used to attempt to establish communication. However, if this network path is unsuccessful as well, there is no communication between SEs. This condition could be caused by network problems, but it is probably due to one of the SEs having hardware problems.

When this loss of communication between SEs occurs, recovery actions are performed to attempt to reestablish communication. However, if this is unsuccessful, the system does an automatic problem report to trigger a service action. In addition, this loss of communication eventually triggers an automatic switchover if checking conditions allow it.

Cage controller boot server controls

Both the primary SE and the alternate SE are used as microcode boot servers for the cage controllers (the internal controllers shown in Figure 1) [4]. The SEs communicate with the cage controllers only through the service network using TCP/IP. It is also important to note that the SEs can be used for loading the cage controller microcode even while in the preSE state (before assigning the primary and alternate SE roles). The reason for this is to provide two servers for quickly loading up to ten cage controllers in the system when the z900 is brought up from a standby power-off state.

This design is quite simple as long as the cage controller microcode loads on the primary and alternate SEs are the same. However, if they are different, the cage controllers must ultimately run with the primary SE cage controller microcode level. The primary and alternate SEs have implemented a set of change management synchronization controls to ensure that cage controllers finally end up with the primary SE cage controller microcode load.

Future investigations

The customer and the service groups have utilized the alternate SE as a quick recovery mechanism in case the primary SE has a problem. In addition, use of the alternate SE as a preload device is also being investigated in order to reduce the amount of system downtime as part of a SE engineering changes task.

The engineering changes task currently takes about one hour to deactivate the system (power-off), load the primary SE with a complete new level of microcode, and then activate the system [power-on/IML (initial microcode load)]. IBM is in the process of investigating a solution which will reduce the time by half by utilizing the alternate SE as a preload device.

z900 alternate SE preload

For z900, an alternate SE preload is being investigated in which the service representative utilizes a series of manual steps in order to cut the downtime in half:

1. The service representative would first do a backup critical data of the primary SE.
2. Next he or she would execute an alternate SE mirror to ensure that both SEs were equal.
3. The service representative would then start the special code load task on the HMC.
4. He or she would invoke the save configuration task on the alternate SE.
5. The service representative would next insert an alternate SE preload diskette into the alternate SE and reboot the SE to load the new SE engineering changes microcode onto the alternate SE hard disk. Again, at the end of this operation, the diskette would force the alternate SE to be disabled as a cage controller microcode boot server.
6. All of the steps up to this point could be performed while the operating system(s) continued to execute on the z900. (The alternate SE preload task would run concurrently with system operations.)
7. Finally, at the customer's convenience, the service representative would perform an alternate SE disruptive switch. This task would deactivate the z900

and cause the preloaded alternate SE to become the new primary SE. The execution of this operation plus the activation (power-on/IML) would take about 30 minutes.

Future processor alternate SE preload

For future systems, IBM is investigating an alternate SE preload task which does not require multiple manual steps, potentially allowing the customer to be able to perform the task as well as including the option to add any Licensed Internal Code changes (patches) released since the SE engineering changes CD was built. Some highlights of this proposal are as follows:

1. The user would first perform a backup critical data of the primary SE.
2. Next, he or she would execute an alternate SE preload task. Instead of having multiple manual steps, the following would automatically be performed internally.
   - An alternate SE mirror would be performed.
   - The alternate SE would have a new engineering changes level of microcode loaded onto its hard disk.
   - A user option would allow Licensed Internal Code changes (patches) to be applied to the preloaded alternate SE hard disk (Licensed Internal Code changes can be applied concurrently with z900 system operations, but when a totally new microcode EC is applied, many customers and service representatives prefer to apply the Licensed Internal Code changes at the same time).
3. Again, the above overall operations would be concurrent with z900 operating system execution.
4. Finally, at the customer's convenience, the user would perform the alternate SE disruptive switch.

Again, the previous one-hour customer downtime has been reduced to 30 minutes. However, if the service representative chose to perform Licensed Internal Code changes in addition to the engineering changes task, this time is also saved by the alternate SE preload. This means that if the service representative previously performed both tasks, the total time would have taken an average of 1½ hours, but using alternate SE preload, it would take only 30 minutes of customer downtime, even though the alternate SE preload time (which is concurrent with system operations) would take 1½ hours.

Conclusions

The alternate SE was originally released as a feature to provide a second SE for quicker recovery time to get another SE operational until the failed SE could be repaired or replaced. Over time, the alternate SE has been enhanced to move its network traffic from the customer network (especially the mirroring data, which can be large) to the service network. An intelligent switch checking function provides support to ensure that the customer will not switch his or her system to a new primary SE which is in a worse state than the previous one. Automatic switchover puts the onus on the Licensed Internal Code to determine when an alternate SE switch is required, and it should be quicker and less error-prone than human observation of the problem, which was the previous G6 design.

The potential future function of alternate SE preload would dramatically reduce costly system downtime during engineering changes for microcode.

The alternate SE provides a rich set of reliability, availability, and serviceability features for the z900 and future zSeries systems, with the alternate SE operation and recovery being concurrent with zSeries system operation.

References

1. C. L. Chen, N. N. Tendolkar, A. J. Sutton, M. Y. Hsiao, and D. C. Bossen, "Fault-Tolerance Design of the IBM Enterprise System/9000 Type 9021 Processors," *IBM J. Res. & Dev.* **36,** No. 4, 765–779 (1992).
2. B. E. Casey, G. L. Dunlap, M. C. Enichen, D. A. Larnerd, J. A. Morrell, S. R. Nicholls, P. D. Pagerey, and S. L. Rockwell, "A Method and System for Providing a Common Hardware System Console Interface in Data Processing Systems," U.S. Patent 6,182,106, 2001.
3. M. Mueller, L. C. Alves, W. Fischer, M. L. Fair, and I. Modi, "RAS Strategy for IBM S/390 G5 and G6," *IBM J. Res. & Dev.* **43,** No. 5/6, 875–887 (1999).
4. F. Baitinger, H. Elfering, G. Kreissig, D. Metz, J. Saalmueller, and F. Scholz, "System Control Structure of the IBM eServer z900," *IBM J. Res. & Dev.* **46,** No. 4/5, 523–535 (2002, this issue).

Received August 27, 2001; accepted for publication April 4, 2002

Brian D. Valentine *IBM Server Group, 1701 North Street, Endicott, New York 13760 (bdvalent@us.ibm.com).* Mr. Valentine is a Senior Programmer working in the HMC (hardware management console) and SE (support element) Licensed Internal Code Development Group. He graduated from Pennsylvania State University in 1983 with a B.S. degree in computer science, joining IBM that same year to work in the 9370 Support Processor Licensed Internal Code Development Group. He has also worked in the Support Processor Licensed Internal Code Development groups for IBM 9371, 9221, CMOS G1 to G6 processors, and the z900.

Helmut Weber *IBM Server Group, Schoenaicherstrasse 220, 71032 Boeblingen, Germany (weberh@de.ibm.com).* Dr. Weber is a Senior Technical Staff Member currently working on IBM eServer system design concepts. He graduated in 1979 from the University of Marburg, Germany, with an M.S. degree in mathematics and physics; he received a Ph.D. degree in mathematics from the University of Marburg in 1982. He joined IBM in 1984 at the Boeblingen Development Laboratory, where he worked on operating system concepts in the Advanced Technology Group. Between 1988 and 1999, Dr. Weber worked on most support processor development projects in the IBM Boeblingen and Endicott laboratories: S/390 systems 9370 and 9371, S/390 CMOS G1 to G6, and the zSeries z900 server. Since 1999, Dr. Weber has been on international assignment, working in the IBM Poughkeepsie laboratory.

John D. Eggleston *IBM Server Group, 1701 North Street, Endicott, New York 13760 (egglestj@us.ibm.com).* Mr. Eggleston is an Advisory Programmer working in the HMC (hardware management console) and SE (support element) Licensed Internal Code Development Group. He graduated from Rensselaer Polytechnic Institute in 1982 with a B.S. degree in mathematics, and from Syracuse University in 1988 with an M.S. degree in computer and information science. He joined IBM in 1982 working on application development for the Endicott integrated circuit line. Mr. Eggleston worked in various development groups before joining IBM S/390 development in 1995.