# Perspectives on the AIN Architecture

History, architecture, and evolution provide three points from which to view the Advanced Intelligent Network.

Roger K. Berman and John H. Brewster

This article will examine the Advanced Intelligent Network (AIN) from a variety of perspectives: historical, present-day architecture, and future evolution.

From a historical perspective, the history of the AIN is traced from predivestiture 800 and calling card service capabilities, through IN/1, IN/2, and IN/1+, leading to the various AIN releases. In addition, motivation for, and results from processes such as Bellcore's multivendor interactions also are discussed.

Following the historical perspective, the present-day view of the AIN architecture is described, including the switching system and other network systems, as well as operations. The AIN functionality supported by this architecture is then described from a customer point of view, by means of an illustrative service which could be provided from an AIN platform. Finally, the next steps in the AIN evolution are discussed.

## Intelligent Network History

During the past 25 years telephone networks have greatly expanded in terms of the number of subscribers served and the volume of traffic carried. An even more dramatic expansion has occurred in the range of services offered to subscribers. The expansion of services has been driven by the needs of the increasingly sophisticated business and residential subscribers, and has been enabled by the widespread deployment of stored program control (SPC) technology in the telephone networks.

The intelligence provided by SPC technology initially was centered in the network switching systems. The first such system, in 1965, was AT&T's No. 1 ESS. Residential services such as Call Waiting, and business services such as Centrex were introduced early in the No. 1 ESS life cycle.

During the 1970s, SPC intelligence was beginning to be introduced in systems whose functions were to support the network's management and maintenance. These systems, collectively known as Operations Systems (OS), were needed in order to deal with the increasing complexity of the network and the services offered to its subscribers. Since their initial introduction, the functions provided by the OSs have expanded to include every aspect of network planning, engineering, administration, and maintenance.

Beginning in 1981, network intelligence reached a new plateau when AT&T introduced the use of centralized databases to support Calling Card and 800 Service. These databases are located at network control point (NCP) systems and are accessed by SPC switches via the Common Channel Interoffice Signaling (CCIS) network. This centralized approach allows the introduction of some services that would otherwise be impractical due to the complexity of managing large amounts of volatile data at every SPC switch.

After divestiture of the Bell System, the newly formed regional companies deployed centralized databases at service control points (SCP) to support alternate billing services (ABS) and 800 calling. ABS provides calling card validation and other line information functions for collect and third-party billing. Access to the SCPs from switches is provided via Signaling System 7 (SS7) networks. The functionality in the switches, SS7 network, SCPs, and OSs that support these services is collectively known as Intelligent Network 1 (IN/1) [1].

Recognizing that there was a potentially large number of services that might be offered by expanding the IN/1 functionality, efforts were undertaken in Bellcore at the request of the regional companies, to define an architecture that could realize that potential. That effort's result was a plan for an architecture known as Intelligent Network/2 (IN/2). This architecture included a greatly expanded set of switch and SCP capabilities, known as functional components (FC), and a new system, called the intelligent peripheral (IP). The FCs defined the atomic functionality elements of the architecture and the IP provided a platform for deploying service-assistance capabilities. This architecture was intended to be capable of supporting a wide range of voice and data services, and, being telco-programmable, would facilitate rapid deployment of those services.

Based upon analyses by the regional companies and switch vendors, it was determined that IN/2 was an overly ambitious proposal which would entail unacceptably high risks and could not be implemented in a sufficiently short time frame. Atten-

*Roger Berman is district manager at Bellcore where he is responsible for the Advanced Intelligent Network (AIN) Project Management and Standards.*

*John Brewster is a district manager at Bellcore where he is responsible for architecture and transition planning for the Advanced Intelligent Network.*



0163-6804/92/$03.00 1992© IEEE

tion was then focused on designing an architecture with a subset of the IN/2 functionality which could introduce service-independent capabilities to the network within a few years. The resulting architecture, known as IN/1+, was described in Bellcore SR-NPL-001052 ("IN/1+ Network Baseline Architecture") in May 1988. Similar to IN/2, it also included the use of FCs and IPs, but they were limited to a small set of functions needed to support voiceband services.

The switch vendor reactions to IN/1+ generally were supportive, but there was a growing concern that an SCP/SS7-based architecture would not be able to provide a sufficient performance level to support all the potentially desired services. There also was a perception that it would be advantageous to achieve better alignment between the IN architecture and the architectures of the network switches. Based upon these factors, it was decided to stop the IN/1+ effort and to convene a forum, called Multi-Vendor Interactions (MVI), encompassing Bellcore, the regional companies, and the vendor community, with the purpose of collecting the best ideas from all these sources to define an architecture that would meet the long-term objectives of the regional companies and have the support of the vendors.

The MVI process was initiated by Bellcore and the regional companies with a notice in the October 1988 issue of the Bellcore Digest [i]inviting participation from telecommunications switching and computer hardware and software vendors who were willing to provide significant resources to the MVI process. Sixteen vendors responded, and the MVI forum was launched at the beginning of 1989. The number of participating vendors grew quickly to 22. The technical results from the MVI process were published in March 1990 [2].

The next stage of the IN evolution begun by MVI was labeled the Advanced Intelligent Network (AIN). An evolutionary path was defined by a series of AIN releases. Each release contains additional architectural attributes and capabilities for supporting services. The initial step is AIN Release 0, which was beginning to be deployed in 1991. It is being implemented with different architectures in accordance with specifications produced independently by different regional companies. These specifications were produced in parallel with the MVI process, and thus were not based on the MVI results.

AIN Release 1, targeted for later deployment, encompasses many of the concepts from the different Release 0 architectures within a single functional architecture. AIN capabilities beyond Release 1 are in an early planning stage. They will evolve from AIN Release 1 and support an expanded range of information networking services. The MVI process addressed both the Release 1 and the post-Release 1 phases of AIN.

The following description of AIN architecture represents the present view of AIN Release 1 [3].

## AIN Architecture

The AIN architecture was derived from a set of functional needs associated with the provision of voiceband telecommunications services. To support these needs, a call model that includes relevant call states and connectivity attributes was developed (see companion article on the AIN call model). The AIN call model is largely based upon results of the MVI call model work.

A primary feature of the AIN architecture is its flexibility regarding its physical realization. The physical systems and interfaces included in the AIN architecture are illustrated in Fig. 1.

Not all the systems identified as components of the AIN architecture, however, need to be deployed to provide AIN service capabilities. The decision concerning which network elements to deploy will depend on a variety of factors, including existing network characteristics and deployment plans, types of services to be deployed, and ubiquity of services to be deployed.

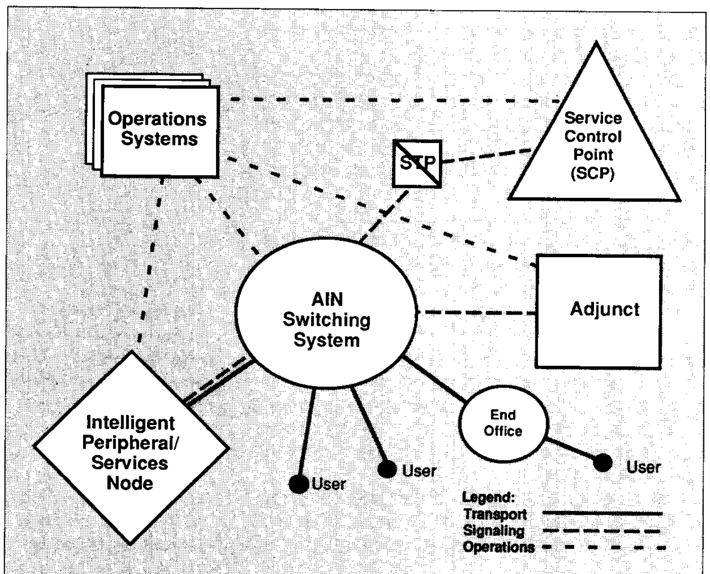
Although individual network implementations of AIN may vary, all share the following characteristics:

AIN services are provided through interactions between switching systems and the systems supporting AIN service logic. The switching system detects conditions for AIN service involvement (i.e., it encounters a "trigger"), formulates an AIN service request, and responds to call processing instructions from the network element in which the AIN service logic resides.

One or more vehicles for providing AIN service logic are available in the network. Depending on implementation considerations and the services to be offered, AIN service logic may be provided at a service control point (SCP), adjunct system, services node, or within a programmable platform provided at the switching system.

Methods exist to allow network interaction with users. The interaction may involve the provision of service-specific announcements to a user or the collection of digits input by a user. The participant interaction resources may reside in a switching system or may be provided by an intelligent peripheral (IP). A services node may be used to provide participant interaction capabilities and AIN service logic in a single network system.

Systems and procedures are in place supporting service negotiations, backup of customer data, trouble detection and recovery, data for traffic engineering, billing, and other operations tasks.




■ Figure 1. *AIN release 1 Architecture*

Each physical system in the AIN architecture shown in Fig. 1 will now be described briefly.

### AIN Switching Systems

The AIN switching system [4] is the hub of the AIN architecture. The AIN switching system identifies calls associated with AIN services, detects when conditions for AIN service involvement are met, formulates requests for call processing instructions from AIN service logic, and responds to the instructions received. The call model provides a framework for formulating these requests and getting responses back from AIN service logic. (The call model is described in more detail in the appendix.)

The AIN switching system capabilities may be deployed in an access tandem, local tandem, or end office. If the AIN switching system is acting as an access tandem or local tandem, it is able to provide limited AIN services to users connected to subtending switching systems.

Switching systems which are not equipped with the AIN switching system capabilities may or may not be equipped with a Network Access Point (NAP) capability. (Although not explicitly shown in Fig. 1, the NAP capability resides in the end office shown.) Those switching systems that are NAP-equipped can provide access to certain AIN originating services for their subtending subscribers. On detecting a request for one of these AIN services, the NAP routes the call, along with the signaling information necessary to identify the calling user and the request for AIN treatment, to the AIN switching system from which it is served.

For those switching systems which are equipped with neither AIN switching system nor network access point capabilities, only AIN calls that can be detected based on existing dialed digits translation or class of service can be routed to, and recognized at, the serving AIN switching system.

### Systems Supporting AIN Service Logic

The SCP [5] invokes Service Logic Programs (SLPs) [6, 7] in response to external messages received from an AIN switching system and internal messages that originate within the SCP. The Common Channel Signaling (CCS) network allows SCPs to be fully interconnected with AIN switching systems through one or more signaling transfer points (STPs). Because of this characteristic, SCPs are well-suited to support network service capabilities such as those required for 800/Freephone, area number calling or person locator services.

The adjunct has a direct communication link to an AIN switching system (as opposed to using the CCS network). Since a high-speed interface is envisioned for use between the adjunct and the AIN switching system, there may be performance considerations which make the adjunct an appropriate system for supporting services requiring quick responses to user actions (e.g., services which control provision of dial tone or which may result in reconfiguration of call connections).

The SCP and adjunct are key systems which provide the AIN programming environment allowing for rapid introduction of new subscriber services. In particular, these systems support the set of functions which define the network capabilities available for use in providing AIN services. These functions are defined in a service-independent manner so they may be used and re-used for a variety of AIN applications, as defined by the SLPs.

The services node provides access to AIN SLPs and, in the near term, the services node may be specialized to support a specific service or set of services rather than supporting the full range of network functions being defined for the SCP and adjunct. The services node communicates directly with an AIN switching system via ISDN access links. The services node, like the IP described below, supports the ability for user interaction to allow collection of dialed digits or spoken input from users, as well as provision of customized announcements to users.

### Participant Interaction Systems

The intelligent peripheral (IP) [8], is a system which controls and manages resources such as voice synthesis, announcements, speech recognition, and digit collection. The AIN switching system routes a call to an IP as necessary to support the request of such resources by AIN service logic. In establishing the call connection to an IP, the AIN switching system also passes message parameters which instruct the IP to perform specific user-interaction functions. When the IP completes the requested functions, it returns any information collected from the user to AIN service logic via the AIN switching system.

In the case of a services node, the identification of the need for specific participant interaction resources and the actions required to provide the user interactions may be provided in a single system.

### Operations Systems

AIN Release 1 network operations functions provide memory administration, surveillance, network testing, network traffic management, and network data collection [9]. These functions support the provisioning, maintenance, and operation of the elements of the AIN Release 1 architecture and the services that are enabled by it. The operations functions are located in the network components of the architecture, such as the switches, and in operations applications (OAs) that reside in the OSs. The functions located in the network components generally are specific to that component, whereas OAs provide functions that transcend a single network component. In many cases, the OAs will be developed as extensions to currently deployed, or replacement OSs. In other cases, however, the OAs may be in new OSs, introduced to support AIN Release 1.

### Interfaces

The interfaces between AIN network elements are selected to support standard protocols: Interfaces between the switching system and SCPs or adjuncts will use the SS7 Transaction Capabilities Application Part (TCAP) as the application layer protocol [10]; ISDN interfaces will be used between the switching system and IPs or services nodes; and standard operations interfaces based on the X.25 protocol will be used for AIN. End users may access AIN services from either conventional analog or ISDN interfaces.

## Service Example

We will now present an example of a service, called Portable Speed Calling List (PSCL), that could potentially be implemented using the AIN Release 1 architecture. It is given as an illustrative example, to show how the elements of the architecture could be used

*The AIN switching system capabilities may be deployed in an access tandem, local tandem, or end office.*

to support services. It is not implied, however, that any of the regional telephone companies are specifically planning to use AIN Release 1 to deploy PSCL.

The PSCL service can be characterized as allowing callers to use speed calling numbers, which may be one- or two-digit codes that represent complete seven- or 10-digit telephone numbers, to make calls from telephones other than their own. Each PSCL subscriber could have his/her own list of codes. To use PSCL, the caller would need to dial an access number, possibly a service code such as *55, enter a personal identification number (PIN), and the speed-calling number. To provide user friendliness, the caller could receive interruptible prompts and acknowledgments at each step.

The flow of events and actions that might take place in an AIN Release 1 implementation of PSCL are as described below. The numbers in the text refer to the communications paths shown on Fig. 2.

First, the caller dials the access code for PSCL (1). The AIN switch detects a trigger based on this access code, which causes a message to be sent to an SCP or adjunct (2) to notify it of the trigger, and to wait for subsequent processing instructions.

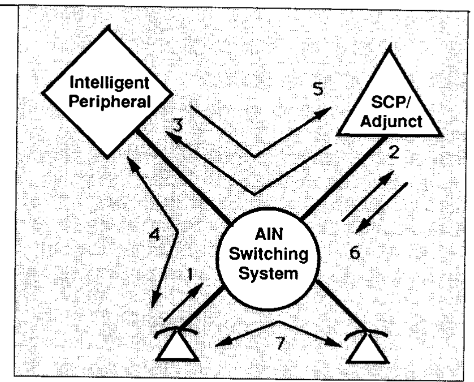
Second, the SCP/adjunct responds to the message from the AIN switch by invoking an SLP for PSCL. The PSCL SLP analyzes the information received from the AIN switch and requests the SCP/adjunct to return a message to the switch (3), instructing it to connect the caller to an intelligent peripheral (IP), and to inform the IP about the resources/procedures to be used on the call.

Third, the IP allocates resources that provide voice prompts to the caller to request that he/she enter their PIN and the speed-calling number (4). Upon receiving the information, the IP returns it in a message (5), via the AIN switch, to the PSCL SLP in the SCP/adjunct.

Lastly, the PSCL SLP validates the caller's PIN and determines the telephone number corresponding to the speed-calling number. The PSCL SLP then requests the SCP/adjunct to send a message to the AIN switch (6) directing it to set up a call from the caller to the requested number (7).

In this simple example, it is shown how the elements of the AIN Release 1 architecture can work together to implement services for telephone network users. Inevitably, an actual service design would be more complex than this example since exception conditions (such as an invalid PIN) and service management functions (such as modification of the list entries) would need to be handled. The reader, however, can easily imagine how the illustrated techniques can be utilized to provide these additional functions.

The concepts behind AIN have been evolving for more than a decade. A great deal of interaction between Bellcore, the regional companies, and vendor participants has embellished these concepts, nurturing them from the speculative idea stage to a set of highly detailed specifications that provide a wide range of call processing and operations functions. We know, however, that Release 1 is not the final chapter of the AIN story. Voiceband service needs that go beyond the AIN Release 1 capabilities are already recognized. For example, to support personal communications services, additional information flows between the AIN architectural elements will be required. Broadband technology will also usher in additional functionality needs that AIN will be called upon to support. The process for managing the evolution of AIN in response to these needs is not yet defined, but it must certainly rely on the continuing cooperation and commitment of all facets of the industry.




■ **Figure 2:** *Illustration of PSCL implementation via the AIN Release 1 architecture*

### References

[1] R. B. Robrock II, "The Intelligent Network—Changing the Face of Telecommunications," Proc. IEEE, Vol. 79, No. 1, pp. 7-20, Jan. 1991.
[2] Bellcore Special Report SR-TSY-001629, "Bellcore Multi-Vendor Interactions Compendium of 1989 Technical Results," Issue 1, March 1990.
[3] Bellcore Special Report SR-NPL-001623, "Advanced Intelligent Network Release 1 Network and Operations Plan," Issue 1, June 1990.
[4] Bellcore Technical Advisory TA-NWT-001123, "Advanced Intelligent Network Release 1 Switching Systems Generic Requirements," Issue 1, May 1991.
[5] Bellcore Technical Advisory TA-NWT-001125, "Advanced Intelligent Network Release 1 Service Control Point Generic Requirements," Issue 1, May 1991.
[6] Bellcore Framework Technical Advisory FA-NWT-001132, "Advanced Intelligent Network Release 1 Service Logic Program Framework Generic Requirements," Issue 1, May 1991.
[7] Bellcore Technical Advisory TA-NWT-001124, "Advanced Intelligent Network Release 1 Service Logic Execution Environment Generic Requirements," Issue 1, May 1991.
[8] Bellcore Technical Advisory TA-NWT-001129, "Advanced Intelligent Network Release 1 Intelligent Peripheral Interface Generic Requirements," Issue 1, September 1991.
[9] Bellcore Technical Advisory TA-NWT-001182, "Advanced Intelligent Network Release 1 Logical Data Model Generic Requirements," Issue 1, May 1991.
[10] Bellcore Technical Advisory TA-NWT-001126, "Advanced Intelligent Network Release 1 Switch-Service Control Point/Adjunct Application Protocol Interface Generic Requirements," Issue 1, May 1991

## Appendix:
# IN Call Model

The purpose of this article, which is a companion to "Perspectives on the AIN Architecture," is to provide more detailed information about the AIN Release 1 call model. It is separated from the main article because it explores details which are not necessary for a basic understanding of AIN, but which cannot be avoided when describing the AIN Release 1 call model.

The AIN Release 1 call model builds on the current call processing infrastructure of existing digital switches. The call model is comprised of two components, the basic call model (BCM) and the connection view (CV). The BCM provides a generic model of current call processing for basic two-party calls, and describes when in call processing AIN switch capabilities can be invoked. The CV describes how service logic residing in the SCP/adjunct can access these AIN switch capabilities to influence call processing in the switch.

### Basic Call Model (BCM)

The BCM generically represents basic call processing in the AIN switch and when, in call processing, service logic residing in the service control point (SCP) or adjunct can receive notification of call processing events (e.g., information collected, route selected, call presented, busy, ringing timeout, etc.), so it can influence subsequent call processing in the AIN switch.

The BCM is comprised of an originating BCM (associated with the calling party) and a terminating BCM (associated with the called party), as shown in Figs. 3 and 4. The points in call (PICs) shown in the figures represent the sequence of actions the AIN switch performs at each particular point in processing a basic two-party call. The trigger check points (TCPs), also shown in the figures, identify where, in the processing of a call, service logic in the SCP/adjunct can receive notification of switch events in order to influence subsequent call processing in the AIN switch.

For example, if the AIN switch detects a trigger, it sends a message to service logic (residing in the SCP/adjunct), which may request the switch to continue processing the call as normal, to resume at a different PIC, or to manage some interactions with the user. At the completion of AIN processing, if the AIN switch then detects that a switch-based feature should be engaged, it will invoke that feature.

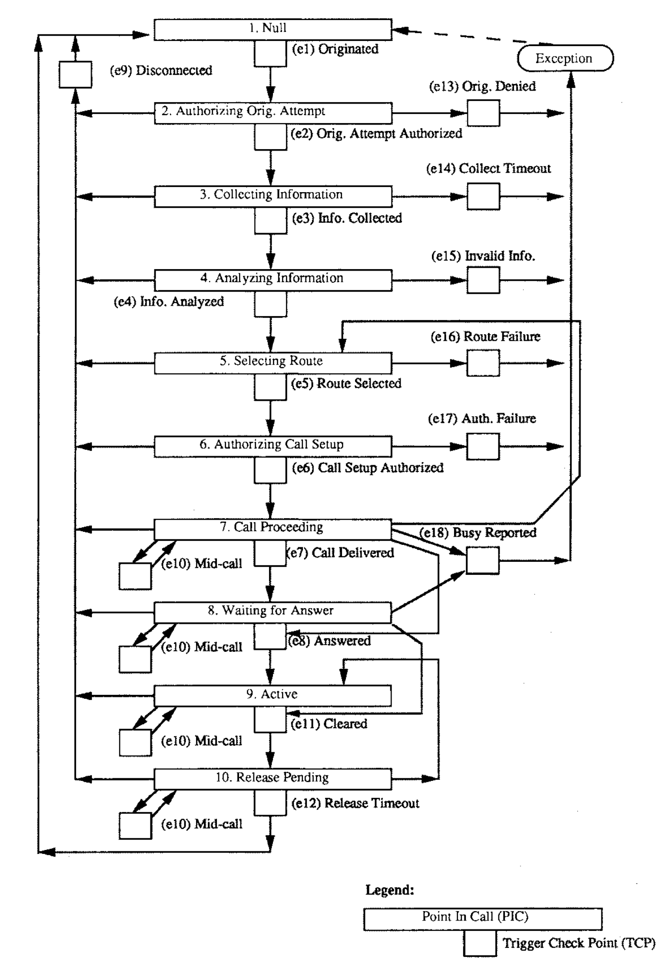
The TCPs that lead to the box labeled "Exception" model the call processing of incomplete calls and allow the SCP/adjunct to request the AIN switch to take special actions. The Exception box represents call processing that results when an exception event such as timeouts or network busy occurs.

In the originating BCM shown in Fig. 3, the PICs model three different stages of call processing: call setup (the first six PICs), stable call (PICs seven through nine), and call clearing (PIC 10). In the terminating BCM shown in Fig. 4, the PICs model the same three stages, with call setup associated with PICs 11 through 14, stable call associated with PICs 15 and 16, and call clearing associated with PIC 17.

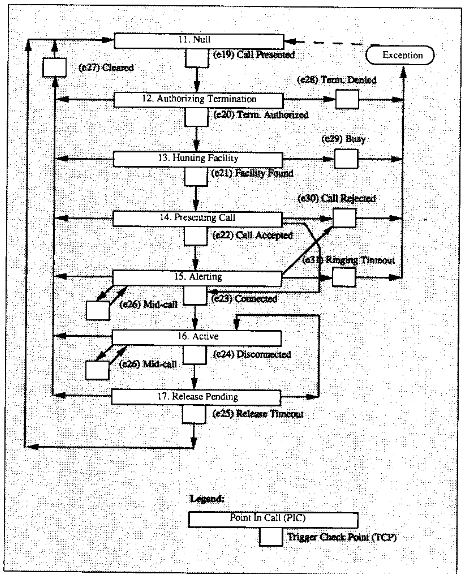
### Connection View

The connection view (see Fig. 5), on the other hand, describes how service logic residing in the SCP/adjunct can access the AIN switch capabilities to influence call processing (as modeled by the BCM). It also provides a generic view of call processing resources in the AIN switch to the SCP/adjunct. This view is independent of switch vendor implementation, and represents the essential characteristics of call processing resources needed by service logic, while hiding their physical details (which may vary by vendor), and their technical complexity.

These switch characteristics include both call processing and connectivity aspects. The call processing aspects include setting up and maintaining the call (as represented by the BCM). They also include the information associated with the call, such as dialed digits, routing information, and billing information. The connectivity aspects include the communication paths to the parties involved in the call, referred to as legs, and the interconnec-




■ **Figure 3.** *Originating BCM*




■ **Figure 4:** *Terminating BCM*

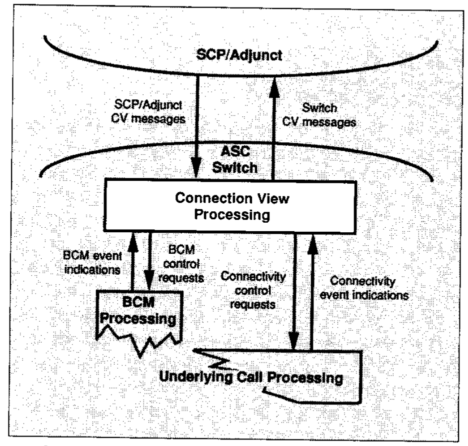
tion of these paths, referred to as a connection point.

The CV provides the ability for service logic in the SCP/adjunct to influence call processing and connectivity aspects in the switch, thereby influencing the flow of call processing (e.g., to provide serial calling or to resume call processing at specific PICs), changing call processing information (e.g., to provide address translation, alternate route selection, or alternate billing), and changing connectivity (e.g., to place parties on hold, to add a third party to the two-party call, or to conference call parties).

The particular information flows related to CV processing are highlighted in Fig. 5. Given these information flows, CV processing translates external service logic instructions (sent by a Service Logic Program in an SCP/adjunct) into operations understood by internal AIN switch call processing. Connection view processing also translates internal call processing events and the state of internal call processing resources into reports that can be understood by external service logic. With this information flow model, CV processing can be described in a generic manner without an actual implementation of AIN switch call processing.

Referring to Fig. 5, the following is a typical information flow scenario:

When a call processing event, such as information collected or route selected (TCPs e3 or e5 in




■ **Figure 5.** *Connection view information flows*

Fig. 3), occurs, BCM processing reports this event to CV processing.

When CV processing receives the indication reporting a switch event, CV processing reports the event and the state of call processing to the Service Logic Program residing in the SCP/adjunct.

Connection view processing in the switch then waits for an SCP/adjunct CV message requesting the AIN switch to perform specific actions.

When CV processing receives the SCP/adjunct CV message, it performs the operation requested in the message by sending BCM and/or connectivity control requests, as appropriate.

When these actions are successfully completed, or fail to complete, BCM processing and/or underlying call processing return BCM and/or connectivity event indications to CV processing, reporting the outcome (either success or failure) of these actions.

Based on these outcome event indications, CV processing then updates the state of call processing, and may report the outcome to the Service Logic Program.

## Biography

Roger Berman received B.Sc. and M.Sc. degrees in electrical engineering from Cornell University, and an M.B.A. degree from New York University. At one time he was responsible for the Bellcore AIN Multi-Vendor Interactions and various AIN generic requirements. Mr. Berman currently is district manager at Bellcore where he is responsible for the Advanced Intelligent Network (AIN) Project Management and Standards.

John Brewster received B.Sc. and M.Sc. degrees in electrical engineering from Yale University and New York University, respectively. He was involved in network planning for the 800 Database and Alternate Billing Services of Intelligent Network/1. Currently, Mr. Brewster is a district manager at Bellcore where he is responsible for architecture and transition planning for the Advanced Intelligent Network.