

White Paper - The Computer Telephony Middleware Challenge

Open Telecommunications Framework® (OTF)

March 2, 1998

When a CT OEM creates a system it needs application software, a host PC, media-processing boards and middleware. Middleware in this context is the software framework for system development. It is the software that lives between the application and the media-processing board. It manages system resources, handles call control and routing, system configuration and security, and so on. Where does the CT OEM get the middleware? The choices, of course, are either make or buy.

If the OEM chooses to develop the media processing boards and the CT framework, it is buying into not just an unending development effort, but perhaps an insurmountable task of developing and maintaining proprietary middleware and media-processing system resources. In the simpler times of single-media single-PC systems this was feasible. But with today's need for extensible client-server integrated-media systems, the investment in capital and lost time-to-market makes this an unjustifiable investment.

The alternative is to purchase the CT media-processing boards. But this also means the OEM puts its product-platform strategy in the hands of the board vendor. Why? Because there are no modular, industry-standard, hardware-independent CT system software products on the market. All CT software systems are sold with integrated support for the vendor's traditional media-processing resource boards: voice boards for Dialogic and NMS; fax board for Brooktrout and Gammalink. It's why they were developed in the first place. Using a third-party voice board with Dialogic's CT Media or NMS's CT Access isn't an option because they are captive technologies. Voice and call control, parameter and resource management are considered the base-level functions to those company's software environments. Additional media, if available from the primary vendor, are add-on functions.

The System Resource Challenge

So if the software framework is bundled with a media-processing or network-access product, must every "technology-resource" vendor develop a CT middleware product? Until recently the answer has been yes, they must. But, how can the developer of a single CT system resource, such as voice, data, text-to-speech, or voice codecs, offer the system developer a comprehensive solution to the OEM's need for a multi-media system platform? Without Commetrex' Open Telecommunications Framework there are few alternatives, and none are very attractive.

One alternative would be to form a technology partnership with a company that already has a middleware software platform. This approach has inherent addressable-market limitations and cedes marketing control to the partner. The second alternative is to develop an integrated-media client-server software system, either proprietary or based on S.100, the only vendor-independent specification available. This represents a sizable investment which can only pay off if third-party media-processing resource vendors develop compatible products.

Is S.100 the Answer?

The S.100 specification has two of the critical features needed to support the development of a modern, integrated-media computer telephony system: a client-server architecture and vendor independence. But, S.100 does not include two capabilities essential to bringing programmable switching to computer telephony: there is no support for multi-vendor signaling, and there is no support for system-wide control of switching. This means S.100 provides no solution for connecting a call received by a network interface from vendor A that should be connected to a station interface from vendor B. And because S.100 is a vendor-independent specification it does not mean a given implementation will be vendor-independent. In fact, indications are they will not be.

OTF -- The Alternative Beyond S.100

There is a better alternative. OTF offers an solution with low investment hurdles, the broadest addressable markets, and strategic control perhaps greater than in-house development (which is often severely constrained by manpower, investment and time-to-market obstacles).

OTF is a Commetrex software product that allows computer telephony OEMs to reduce in-house development investment and still maintain control of their strategic product platform. As a vendor-neutral S.100-compliant kernel, OTF can be extended by the OEM or third-party developers to support multi-vendor system resources and client APIs. Client APIs can be both S.100 compliant and proprietary. Plus, OTF includes a comprehensive, open-architecture signaling protocol which allows the OEM to easily implement complex switching functionality, even when system resources are furnished by different vendors. OTF is finally opening computer telephony's efficient value-adding market structure to complex switching applications.

Is this different and better for the CT OEM? The short answer is yes. But to truly understand how and why, it's necessary to compare what S.100 offers against what the industry's OEM's need to move to a higher level of media integration, system size, switching functionality, and vendor independence. Let's look at the needs of the CT OEM in this age of client-server architectures and media and function integration.

In years past, the only way an OEM could control its platform was to divert resources away from its primary value-creation mission to develop the underlying platform. Of course, prior to the availability of value-adding multi-line voice boards an OEM had no choice but to develop a proprietary system platform. That's why the messaging-system companies that entered the market before 1984 (VBX, Octel, Centigram, etc.) all had to develop their own. But with the market demanding integrated-media solutions, the scope of the technologies is simply too great today for "home-grown" solutions. But how does the OEM bring the full gamut of media and switching technologies to bear on its system, yet still maintain control of its strategic platform? The answer is through standardized value-adding interfaces.

Ideally, today's OEM will have the following strategic advantages:

- Control of strategic platform
- Multi-server multi-client system configurability/architecture
- Multi-vendor programmable switching
- Best-of-breed integrated media
- Best-of-breed hardware

But today's market presents a far-from-ideal situation for the OEM. Every vendor of a multi-line system resource board, be it voice, fax, speech recognition, or network interface, requires that the OEM purchase a proprietary software environment to access the board's functionality. So the system developer is forced to master the complexities of the software environment of multiple vendors -- all on the same system. Applications from multiple vendors cooperating on the same system is a distant reality. True client-server systems have yet to be deployed. What's more, the distributed-switching model of MVIP, SCbus, and H.100 makes realizing the ease-of-use of the products of the "programmable switch" vendors only a hoped-for ideal in the CT "space".

It doesn't have to be this way. Two things are needed: a standard specification for a computer telephony operating environment, and a vendor-neutral implementation. The ECTF's S.100 specification is a good step toward the first; Commetrex' OTF adds the second. Moreover, OTF goes beyond simply offering a vendor-neutral implementation of the S.100 kernel. It adds a comprehensive signaling protocol and a centralized connection management and call routing facility which makes developing complex switching systems, such as PBXs and ACDs, economically feasible.

Inside S.100

S.100 is a strong foundation on which an effective client-server CT system can be built. S.100's major components are kernel services and their client APIs, the client APIs for "technology resources", and interfaces to the telephony service providers (S.300). There is one product available today, CT Media from Dialogic, which is based on S.100 (perhaps, visa versa). It includes the technology-

resource APIs and, possibly more important, it also includes the Dialogic voice-board products as fully integrated resources. So, although it may be based on S.100, it's hardly vendor-neutral and definitely not "open". (Try to substitute a different voice board.)

Another issue has to do with the availability (or lack thereof) of S.300. It's not yet available, and until it is made available by the ECTF any ability to achieve vendor-independence at the resource-board level has to be somewhat suspect. Commetrex' approach is to furnish a temporary S.300 substitute and replace it with the S.300 interface when it's available.

Another interface needed for a truly modular system is between the client-level technology-resource APIs and the system kernel. Here, the ECTF has published the S.200 protocol interface. It separates the client API from the kernel while it supports remoting the client API through a telecommunications transport such as TCP/IP. The specification also allows for proprietary "Application Interface Adapters" which can be somewhat simpler than the S.200 implementation.

S.100 Connection Management

But S.100 suffers from the distributed-switching model of all the widely accepted PCM highways (MVIP, H.100, etc.) in its ability to deliver a robust and comprehensive connection-management facility. Because each vendor's resource board may contain a switch, each board represents a significant complicating factor as well as a potential source of an unintended switch setting. This arrangement presents the need for a hierarchy of S.100 System Call Routers (SCRs). The SCR, an optional S.100 system element, shields applications from the S.100 Connection and Group Managers. It normalizes vendor-specific call-control to S.100 functions such as answer call, make call, and disconnect call. But since S.100 does not define a signaling protocol, the process of connecting a call from the network-access interface of Vendor A, for example, to the station interface of Vendor C is extremely complex. Either a rather complex application must be developed which is vendor aware and uses the S.100 Connection and Group Managers, or the SCR must be used and its functional limitations accepted. And it becomes even more complicated. Before a media stream can be used by an application it must be presented to the S.100 Group Manager as a Call Channel Resource (CCR) through an undefined interface. So there are additional interfaces to each vendor-specific line control entity.

The OTF solution to this complicated puzzle is to create a CCR proxy and define a protocol for each resource to communicate with the proxy. Think of the proxy as a central office and the resource as terminal equipment or a subscriber line. The protocol, then, is a telephony signaling protocol. This notion can be extended to any OTF addressable entity, be it a host-based application or a resource-board-based application. If the application does not reside on the board, or entirely on the board, a host-resident proxy can do the job.

In the traditional model, the switch configuration is controlled through vendor-specific switch control and signaling. For example, a line interface from one vendor may use channel-associated signaling while another uses a host-based call-control protocol. In a basic S.100 system each vendor presents significant complications, forcing the OEM towards a single-vendor approach.

But what if the switch fabric were controlled by one connection manager in a manner similar to a PBX? From the system's perspective, calls would originate at the trunk interface. Call setup would then occur between the interface (or its host-based proxy) and the OTF Connection Manager. Once the connection establishment has progressed to the "call proceeding" state, the media stream is made available to the S.100 kernel as a Call Channel Resource (CCR).

OTF furnishes a host-based signaling proxy and API to simplify development of OTF-compatible resources. But a far better approach, and one slated for future releases of OTF, is for the intelligent line card to host the protocol engine and signal the OTF Connection Manager through the H.100 pin reserved for the purpose.

OTF Addressing Framework

All entities -- applications and network-connected entities -- known to the OTF system are addressable by the system and any other OTF Addressable Entity (OAE) for which permission has been granted. OAEs place and accept calls in a manner similar to a terminal on a network.

The OTF Developer's Kit includes the OTF Addressing API, and each API that requires OTF signaling is packaged with it. The developer can use the API to implement OTF signaling support directly on a resource board or in a host-resident proxy. The latter approach allows the OEM to integrate existing boards from any vendor into the OTF Addressing Framework.

Typically, the OTF Connection Manager (OCM) is able to resolve most connection requests by matching the OAE's address with the name of a Group owner listed in its Connection Map. However, there will be situations where OCM has insufficient information to resolve the connection request, such as a "dialed number" in a PBX application. The OTF Addressing Framework provides for such occurrences by allowing OCM to pass the request off to an OAE registered to resolve the connection requests of either a single address or an address group previously defined to OCM. These OAEs are known as OCM "Controlling Entities".

OCM supports the following types of addressing:

- Explicit
- Address Group (not S.100 Group)
- Dialed Number

- Implicit
- Service Requests

In Explicit Addressing the calling OAE provides OCM with the OTF Address of the called entity. For example, a System Kernel service would use explicit addressing to send an unsolicited event to an application.

OCM treats Address Group Addressing in a manner similar to Explicit Addressing. (Here, the Address Group is a homogeneous group of OAEs, not the S.100 Group). In Group Addressing, OCM must first resolve the Group Address to a specific OAE Address. This is accomplished by OCM placing a query to the OTF Resource Manager (RM), passing in the Group Name. The RM will then translate the name into an explicit address via tables, hunting algorithms, etc. The resulting session request is then processed as an explicit addressing request.

In Implicit Addressing the calling entity furnishes no "to" address at all. If a Controlling Process is registered for that calling entity the CM will simply pass the request there. If no Controlling Entity is registered the request will be denied.

Service Requests differ from the other session requests because the calling OAE does not specify in any way the address of the called party. Instead, it requests some service from the network (the OTF Kernel). This is the mechanism used to implement the S.100 Resource APIs.

OTF includes a facility similar to an Intelligent Network Service Control Point (SCP). The OTF Connection Manager (CM), upon detecting the Service Request will communicate the request to the OTF SCP which will attempt to locate a service provider within the OTF system. The SCP will then return the service provider's OTF Address to the CM, which is then able to complete the call.

OTF System Services

OTF includes the system services defined in S.100 that constitute the core services of the system. They include:

- Session/Event Manager,
- Group Manager,
- Connection Manager,
- Container Manager,
- System Call Router,
- Key Value Sets and
- S.100 Session/Event Management

The S.100 Session Management API is functionally implemented by the OTF Session Manager. The OTF Session Manager is used by all OTF-Addressable Entities (OAEs) to register with OCM. Once registered, an OAE can be

communicated with by any other OAE, either explicitly by name or indirectly through group names or implicitly by a resource request. Once registered, all OAEs have a "nailed-up" signaling connection with OCM to support command-event signaling.

S.100 Group Manager

An S.100 "Group" is an object that presents a unified interface for allocation, configuration, interconnection, and hand-off between applications of the resources needed to perform computer telephony functions. The resource that represents a media stream is the "Call Channel Resource" (CCR). An example of a group for a play-message function is a CCR, a "player", and a "Signal Detector" to detect touch tones. Groups may be explicitly reserved by an application or they may be implicitly reserved by invoking a higher-level function.

The system resource used to abstract a switch-fabric connection is the Switch Port (SP). Each Group has an SP as a virtual resource. The S.100 Connection Manager allows an application to make connections between SPs.

S.100 Connection Manager

The S.100 Connection Manager and its API give the application an easy-to-use facility to switch calls from one party to another, to monitor two other parties, and to control a Conferencing Resource, if one is available. The S.100 Connection Manager delegates connection requests to the OTF Connection Manager for execution.

S.100 Container Manager

"The S.100 Container Manager (and its API) provides an operating system-independent mechanism for the storage and interchange of system data between system services, resources, and the application." A "container" is an object made up of data, usually media data, and a set of attributes. The S.100 Container Manager extends the functionality usually found in file systems with features required by Resources to manipulate media data in a convenient manner.

S.100 System Call Router

The System Call Router (SCR) shields the application from the details of the Connection and Group Managers and the need to be aware of any vendor-specific call control. It provides API functions such as answer, make, and disconnect call, greatly simplifying the routine placement of outbound call and accepting inbound calls in traditional CT applications.

S.100 Key Value Set API

S.100 defines a Key Value Set (KV Set) as a collection of Key Value Pairs, where a KV Pair consists of a Key, which is a type of CTSymbol, and a Value, which is a previously defined data type. A Symbol is a constant whose value is controlled and unique across an S.100 system. Symbols defined by the ECTF are known as CTSymbols. CTSymbols have been defined for the Resource APIs

specified by S.100. So a KV Set can be thought of as a parameter name-value pair, and the KV Set API as a kind of parameter-management API. OTF fully supports the KV API.

Open Telecommunications Framework

It is Commetrex' objective to play a major role in the CT industry's attempt to improve its value-creation capacity by offering products which broaden the system developer's choices and improve technology through specialization based on standardized value-adding interfaces.

With Open Telecommunications Framework the CT developer is positioned to better meet the market demand for media-diversity and scaleable architectures by taking maximum advantage of the efforts of others.