Introduction

Connection Management is a software system that is designed for use within Jtec products. The purpose of this document is to describe Connection Management, so as to impart an understanding of exactly what it is, where it is positioned with respect to other products, how its architecture has been designed, how its operates, and various other development issues.

Overview

Connection Management exists to provide connection and routing services for the particular Jtec product that it is operating within. Predominantly, these services involve routing switched calls according to configuration placed into the system by users, but any type of service may exist.

The connections are ultimately made between physical (e.g. ISDN, X.21 ports) or software (e.g. Routing, Monitor tasks) end points in the system.

Product issues

The purpose of this section is to describe where Connection Management fits in relation to other Jtec products. These issues need to be addresses because a major requirement for Connection Management is that it is common across various products.

There are a number of issues that affect the design of the architecture:
1. Connection Management needs to interface with existing and future products. This requires either a common interface provided by other products, or a set of product specific interfaces provided by Connection Management.
2. Connection Management is expected to have a lifetime of 5 to 10 years, or more. This would tend to indicate that the design must take this into account to some extent.
3. As this is a system that will potentially have many concurrent developers, it should be partitioned so that modules can be worked upon separately and in isolation.

Some of the issues that are expected to arise during the lifetime of the system are:
1. New forms of management -- The current form of management is SNMPv1; it is plausible that new forms of management will include such things as SNMPv2, CMIP, Computer-Telephony Interfaces, Command Line Interfaces, etc. Each of these should consist of a new module that is implemented with minimal changes to existing modules.
2. New ports --
3. New calls and components --

Software Architecture

The software architecture is used to describe the high level organisation of Connection Management. This organisation is carried out in such a way to meet several important requirements.

The need for a well designed architecture is evidenced through the following quotes.

> If a project has not achieved a system architecture, including its rationale, the project should not proceed to full-scale system development. Specifying the architecture as a deliverable enables its use throughout the development and maintenance process.
> - Barry Boehm , 1995

> I am more convinced than ever. Conceptual integrity is central to product quality. Having a system architect is the most important single step toward conceptual integrity... After teaching a software engineering laboratory more than 20 times, I came to insist that student teams as small as four people choose a manager, and a separate architect.

- Fred Brooks, The Mythical Man-Month (20th Anniversary Edition). 1995

The Call Management architecture attempts to address a number of issues which are summarised in the following list. It is intended that these be kept in mind across the life of the product to maintain the original integrity of the design.

1. Modules can be tested in isolation. This provides a clearer way to diagnose the location of faults, and to increase confidence in the operation of modules.

2. Modules encapsulate technologies. This means that specific networking or protocol technologies are encapsulated in such a way that they can be replaced with minimal disturbance to other modules.

3. Modules have well defined interfaces. This ensures that modules can be developed in isolation, at the expense of greater design/maintenance effort to retain these interfaces; however the internal complexity of modules is generally reduced.

4. Modules seperate physical and logical aspects of the design. This attempts to reduce problems that may be associated with transportation to other platforms and environments; or changes at the boundaries of the system.

5. Modules can reside on separate physical platforms. This allows for modules at the highest level of control to operate on, say, an external workstation; and the lowest level of control could be distributed across several physical modules.

There are other considerations imposed on the architecture, these manifest themselves as constraints in the following manner.

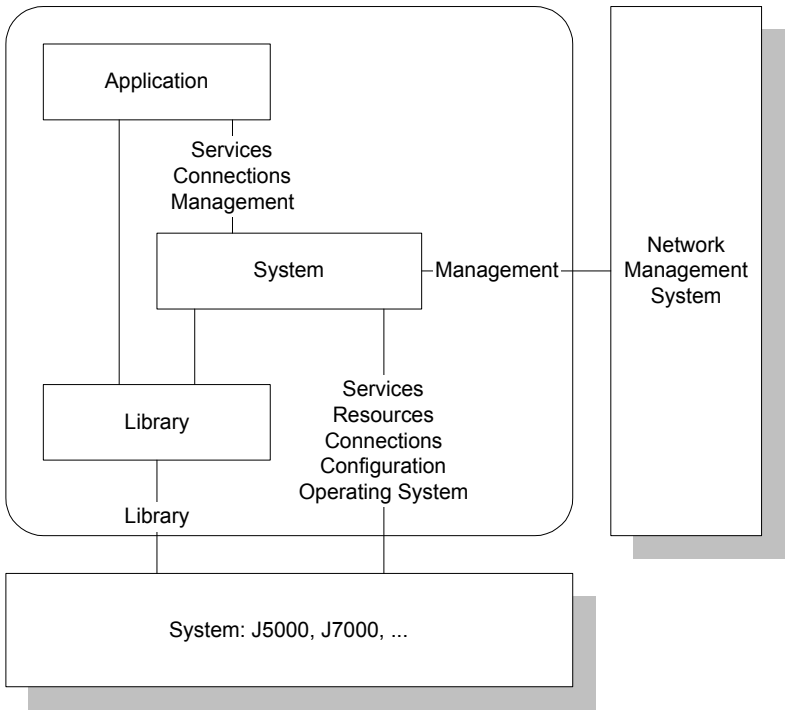
1. The interfaces between internal modules utilise C++ classes. This means that irrespective of the technology employed within a particular module, it must use C++ to communicate with other internal modules.

2. The interfaces to external modules generally utilise JEXEC messages. These interfaces are wrapped into C++ classes so as to hedge against changes to the operating system, but in general this provides the portability across Jtec products.

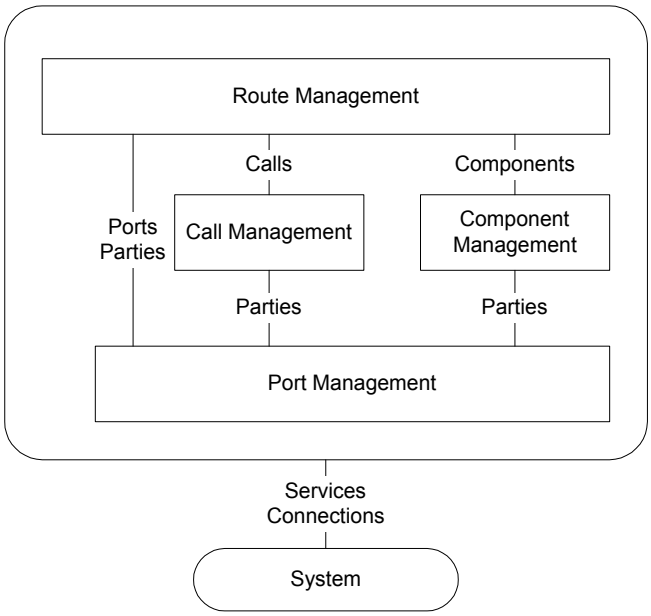
3. The target environment is generally an embedded sytem. This means that code/data size, and execution time present constraints that must be factored into the design. It is considered that in the longer term, design flexibility will be a greater issue as platform costs reduce.

The Call Management architecture is described in the following diagram. The diagram illustrates the major modules, and the interconnections between those modules. Each module may internally contain any type of software in order to fulfill its required functionality.

The first partition involves an Application, System and Library modules. The Application module is further partitioned to contain the Call Management software.



The following diagram illustrates the internal organisation of the Application module.



The major modules in the architecture are described as follows.

1. Management -- This module interfaces to the network management system. It provides the mechanisms to support SNMP operation. A set of C++ base classes provide an interface to the J5000 SNMP agent (Routerware). Using this base, specialised SNMP Items and Tables are constructed, which interact with other modules (e.g. Port Management). Although currently using SNMP technology, the intention of the design is that a similar module based upon CMIP or a CTI standard could be used without significant changes to other modules. These various management modules could potentially co-exist.

2. System -- This module provides the interface to the system elements provided within the product itself. It does this by utilising C++ classes that access system services through direct function call, or

inter-process message interfaces. These mechanisms help to isolate various disparate interfacing issues from a relatively clearn internal interface provided to other Connection Management modules. This is a direct attemp to mitigate portability issues.
3. Support Libraries -- The support library module is responsible for providing various utilities that are used by other Connection Management modules. Most of these utilities are not system specific, and as such, there are minimal portability concerns. Some of these utilities may be constructed by a third party.
4. Port Management -- This collection of modules implements management for each type of port, or endpoint, in the system. These endpoints are terminators or initiators of calls, and may be associated with an actual physical port (e.g. ISDN) or an internal entity (e.g. Router). The modules may translate signalling information and will handle local issues such as tone generation/provision, etc. The interrface to the system is through the system module.
5. Call/Component Management -- These modules are responsible for providing calls and components, that interconnect and connect with parties eminating from port management. This includes such things as point to point calls, signalling translation, voice compression, and so on.
6. Route Management -- This module co-ordinates the activities of ports, calls and components according to configuration specified by the user. It provides the highest level of intelligent decision logic in the system.

Lifecycle/Development Issues

There are a number of technologies used throughout the various stages of development.
1. Booch / Object Oriented Design -- Call Management is designed using Object Oriented techniques, particularily aspects of the Booch methodology. This involves classes, inheritence structures and polymorphic behaviour.
2. Rational ROSE -- The Rational ROSE tool is used to carry out the design, maintain the design, and to generate C++ source code for implementation. This supports class categories, subsystems, configuration/version control and multi-user operation.
3. C++ language -- The C++ language is the target for all software. Initial source code is generated by Rational ROSE, and additional code is then inserted by hand. Compilation is carried out by the Greenhills compiler.
4. Simple Network Management Protocol -- User management is carried out by using SNMP through the Routerware SNMP agent.
5. Design Patterns -- Where possible, existing and new design patterns are used. The purpose of this is to maintain some level of consistency through the system, to assist maintenance (and design!).
6. Concurrent Versions System (CVS) -- Both design information and source code (that generated, and that filled in) are managed using CVS. This does allow for shared use and development.
7. Object Oriented Test Software -- Testing of modules is carried out by using object oriented test code.
8. EMUJEXEC Test Vectors -- To exercise the complete system, as it would fit within a Jtec product under the JEXEC schedular, test vectors are used with EMUJEXEC.

There are a number of standards that have been employed.
1. Q.931 Signalling --
2. SNMPv1 --

There is an attempt to design with re-use in mind. This occurs by using inheritence within the design, to minimise duplication of common mechanisms. In addition to this, class categories and libraries are designed to separate areas that will be common or different across products. This provides some re-use.