RadiSys®

WHITEPAPER

ATCA 4.0 PLATFORM MANAGEMENT

VERSION 1.0 | JANUARY 2011

CONTENTS

As everyone who uses Advanced TCA (ATCA) as a platform for delivering their applications knows, ATCA is an intricate system made of many building blocks, including complex hardware components. These hardware components in turn contain many programmable and firmware components. Over time these programmable devices and firmware components must be provided with in-service field updates to provide new functionality and fixes. Since ATCA platforms are deployed in environments that cannot tolerate lengthy downtime periods, any in-service updates and upgrade procedures must be efficient. If the module or system must be manually updated it can be a time consuming and potentially error prone exercise due to the dependencies between the programmable devices, firmware, and hardware components across the system.

This has not always been the case. The HPI specification from the Service Availability Forum (SAForum) has been adopted by the industry as the interface between application software and the hardware platform. HPI provides APIs for hardware monitoring, control, inventory, hot-swap, firmware upgrades, and diagnostics. The current ATCA model is to use a single HPI implementation on the Shelf Manager to manage the ATCA system through IPMI infrastructure. This provides limited access to some functions such as firmware upgrades and diagnostics.

Building on established leadership in application-ready ATCA platforms and the release of ATCA 4.0 RadiSys is evolving the hardware platform management infrastructure to expand HPI and extend capabilities to provide more uniform platform upgrades. This will involve the introduction of several enhancements to the platform management framework to provide a more extensible, reliable, and customizable approach to managing platform hardware. This includes the distribution of shelf management capabilities onto each blade instead of a single Shelf Manager instance which will provide a much more rich set of hardware platform management capabilities that will be accessible through the central HPI. Further, the introduction of Software Management Framework will greatly reduce the effort and complexity required to upgrade software and firmware in the platform.

The Service Availability Forum is an industry group whose goal is to provide, promote, and facilitate the adoption of high availability and management software interface specifications. This fosters an ecosystem that provides COTS building blocks for the building of high availability network infrastructure products, systems, and services.

These specifications include:

Application Interface Specification (AIS) – provides an interface between the application and the high availability middleware.

Hardware Platform Interface (HPI) – defines the interface between the hardware and the high availability middleware and makes each independent of the other.

For more information, see www.saforum.org

DISTRIBUTED HPI FRAMEWORK

The HPI specification continues to evolve to provide additional platform management capabilities. Examples of new functionality include Firmware Upgrade Management Instruments (FUMIs) and Diagnostic Initiator Management Instruments (DIMIs). Taking an approach that distributes this functionality across the platform effectively creates an agent that manages other agents.

The advantages of a distributed HPI architecture are:

- It expands the capabilities of HPI beyond IPMI (monitoring and a little control) to enable mapping of Management Instruments (MI) such as sensors, controls, and configuration information to hardware.
- It enables the use of DIMI and FUMI APIs for handling remote diagnostics and firmware upgrade while the previous model can only support POST, and IPMC (IPMI Controller) upgrades through the IPMI bus.
- It reduces the burden on the IPMI bus since resources can be accessed through an alternate LAN interface. This also provides an extra layer of reliability for better fault detection since the Shelf Manager can communicate over the LAN interface to the LMP in the case of an IPMI bus fault.
- Its extensible, plug-in architecture enables customers to add new instruments to meet their needs via custom plug-ins.

EXISTING PLATFORM MANAGEMENT ARCHITECTURE

In the early days of ATCA deployment each type of board, as well as each module on each board (such as the IPMI controller or "IPMC", LMP, and U-Boot) in a system had its own set of upgrade tools. With this approach, upgrades required significant time and work since each individual module to be upgraded had to be done individually. Any sort of centralized upgrade management approach required significant effort by the customer.

Later, new tools were introduced to put a "wrapper" around the individual IPMC, LMP, and U-Boot tools to streamline the update of an *individual* blade. Since each board still required individual attention an additional tool was also introduced which provided the ability to drive centralized system upgrades. This reduced the effort required to upgrade an entire system, but functionality was limited and could not be expanded to include customer-specific functionality.

Figure 1 illustrates the existing platform management architecture. In this model, system level information is collected via the HPI interface on the Shelf Management, which uses the IPMI interface to gather information from other blades in the system. Blade management, including software and firmware upgrades, is accomplished via individual interfaces between the System Manager and each blade using SNMP, SSH scripts, and application-specific APIs. Application information can also be gathered from blades using these same APIs.

This platform management architecture does provide the capability for the System Manager to pull together all system information in a single module. However, given the various interfaces that must be used it requires a high degree of integration by the application developer. Further, activities such as blade software and firmware upgrades can be slow and labor intensive.

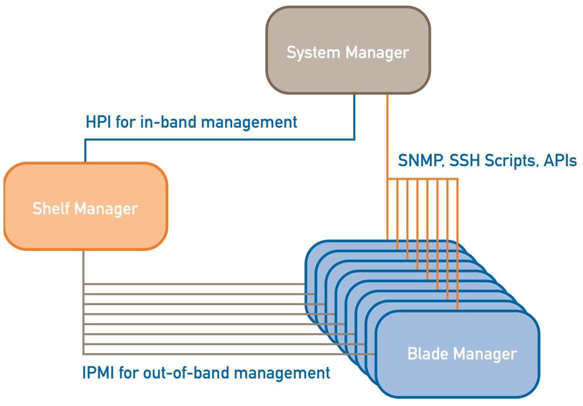

Figure 1. *Existing Platform Management Architecture*

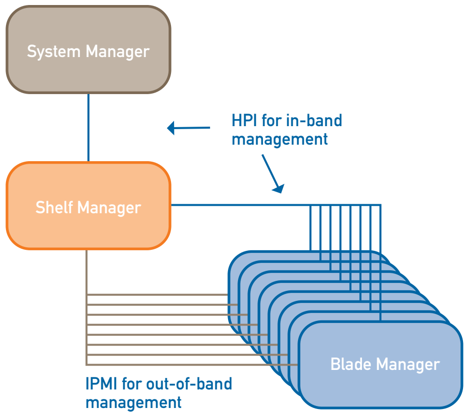

Figure 2. *ATCA 4.0 Platform Management Architecture*

ACTA 4.0 PLATFORM MANAGEMENT ARCHITECTURE

Figure 2 illustrates the platform management architecture that has been introduced with the ATCA 4.0 release. This model greatly simplifies platform management integration since HPI functionality has been expanded to provide all blade management functions, creating a multi-layer architecture and hiding all board-specific functions. This eliminates the necessity for the System Manager to have a discrete interface to each blade in the system.

This greatly decreases the integration effort required for customer applications and middleware. Further, this architecture provides an extensible framework which allows RadiSys and customers to add additional capabilities more simply.

SOFTWARE MANAGEMENT FRAMEWORK

With ATCA 4.0, RadiSys is introducing the Software Management Framework, a set of tools that greatly improves the process of upgrading platform software. The Software Management Framework includes Software Management Library (SML) and the *"rsys_swm"* tool, which leverage the modular HPI FUMI API capabilities.

FIRMWARE UPGRADE MANAGEMENT INSTRUMENTS

As mentioned earlier, new functionality defined in the HPI specification includes Firmware Upgrade Management Instruments, or FUMIs. The purpose of FUMIs is to abstract the firmware upgrade operation and expose it, allowing a remote "C" API to be used to perform the upgrade. FUMIs provide the control structure and API to perform upgrades using different mechanisms, allowing customers to build higher level upgrade applications and simplifying the upgrade process. RadiSys has implemented two types of FUMIs:

- Blade HPI Server for software entities on a module (and its subsidiary modules) which is incorporated in the HPI implementation that runs on the LMP of the hardware module.
- Shelf HPI Server for shelf devices such as fan firmware and PEM firmware which is incorporated in the NPI implementation in the Shelf Manager.

"FUMI Banks" are created that contain the upgrade information. These FUMI Banks are then organized into a "FUMI Bundle" which contains all the FUMIs for a particular module.

SOFTWARE MANAGEMENT LIBRARY (SML)

As already discussed, FUMIs greatly reduce the effort required to upgrade a system by abstracting the process of upgrading entities on hardware modules in the platform to a standard API. However, to further simplify the upgrade process it is desirable to bring these APIs together into a single interface. In some cases, customers may have created a System Manager to coordinate the upgrade process for all entities in a platform and integrate with the RadiSys HPI. However, if this application does not yet exist, RadSys is now providing the Software Management Library, or SML, which is a library that can be used to implement the logic of a System Manager. SML provides a system-level view of the software upgrades with the objective of streamlining the upgrade process. It is an HPI user application which interacts with the Shelf HPI server to manage the upgrade of the shelf or interacts with the Blade HPI Server to manage the upgrade of a blade and its subsidiary modules.

Once a customer receives an upgrade CD from RadiSys, the existing software load is compared to the software on the CD and a campaign can be created to upgrade the managed objects on each individual board. SML includes an API called "SmlInfoGet", which can be used to retrieve the current versions of software entities installed on any of the modules it finds in the platform in a given path. An upgrade API, called "SmlUpgrade" is also included which is used to orchestrate an upgrade campaign on the system. It is possible to create exclusion lists and to implement the upgrade campaign in phases. When an upgrade is done in phases, a specific group of modules is completely upgraded and activated before the next group of modules is upgraded.

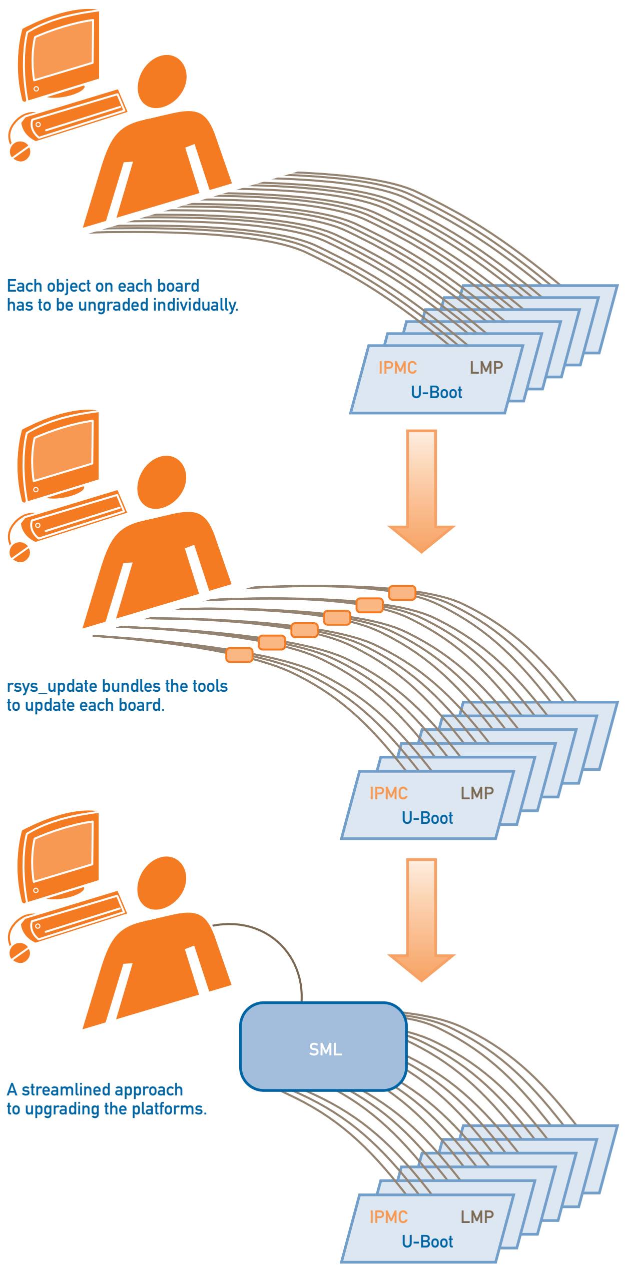
Each object on each board has to be ungraded individually.
IPMC
LMP
U-Boot
rsys_update bundles the tools to update each board.
IPMC
LMP
U-Boot
SML
A streamlined approach to upgrading the platforms.
IPMC
LMP
U-Boot

Figure 3.

The API also supports a callback function which can issue a report during the upgrade detailing current progress and any errors that have been encountered. If an error has occurred the customer will be given the option of changing the action to take if an error occurs. The callback function can also be used to pause the upgrade.

Once complete, a structured report will be issued which will contain the list of modules found and the list of entities that were upgraded during each of the phases, the time it took to perform each stage of the process, and a log of events that occurred during the campaign.

"RSYS-SWM"

If the customer has developed its own System Manager, SML can be integrated directly into it. If not, RadiSys is also providing a sample upgrade utility known as *rsys_swm* which can be used as a starting point to build a custom upgrade application. This is a fully featured utility than can be used to invoke SML operations through the CLI at the shelf or module level. This provides a thin wrapper over SML and has options to show the versions of the firmware and the upgrade of a module or groups of modules in a shelf. The default behavior of rsys-swm is to upgrade all entities under the default root entity path, but additional control over the order and phasing of the upgrade can be controlled through the optional upgrade campaign files.

CONCLUSION

With the ATCA 4.0 release and the introduction of Modular HPI and the Software Management Framework, RadiSys has greatly increased the manageability and upgradeability of ATCA. Now customers can much more quickly and precisely upgrade their field-deployed systems through a simplified interface. ///

RadiSys.

Corporate Headquarters
5445 NE Dawson Creek Drive
Hillsboro, OR 97124 USA
Phone: 503-615-1100
Fax: 503-615-1121
Toll-Free: 800-950-0044
www.radisys.com
info@radisys.com