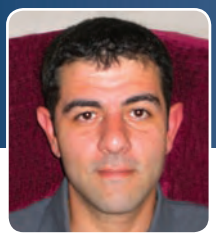

By Tariq Shureih

A closer look at HPI

HPI in a nutshell

The Service Availability Forum's (SA Forum) Hardware Platform Interface (HPI) specification provides a programmatic API aimed at (1) simplifying the development of system management applications and (2) separating the hardware from the management middleware.

HPI derived most of its concepts from the Intelligent Platform Management Interface (IPMI), which defines a platform independent set of capabilities and data types to manage most any computing system. However, HPI provides an additional layer of abstraction that allows for more generic and broader modeling given the platform has enough management capabilities to be modeled.

An HPI system model comprises domains, sessions, resources, and entities (Figure 1). Domains are containers of resources and entities, and sessions are the basic access method to a system (via the domain container).

Resources and entities are the core data and management access points in systems managed and modeled with HPI. Sensors, control capabilities, status indicators, and inventory data are all associated with a resource in a given domain. HPI provides a complete API to access and control these resources in addition to hot swap, managed hot swap, and chassis power controls. With HPI, you have a standardized API that addresses the heart of system management allowing for quick development cycles and portability.

Open HPI (and it is free)

Open HPI is the Open Source implementation of the SA Forum's HPI. The project has gained much popularity in recent times due to the great success in building a solid HPI implementation based on a modular architecture and bundled support for most popular platform management technologies such as IPMI, Simple Network Management Protocol (SNMP), and less intelligent devices exposed via Linux's SYSFS.

Open HPI was designed from the get-go to be a ready-for-commercial-use open source implementation of HPI. The project is licensed under the revised Berkeley Software Distribution (BSD) license, which allows for reuse and derivatives to the code without the mandate to release the derivative code. Further, the open source can be used with proprietary code without contamination.

After two years of development, Open HPI reached high levels of stability, maturity, and experience elevating it to be one of the leading HPI enablers. That is because Open HPI includes:

- Core library (infrastructure)
  - The core C HPI API shared library providing all HPI interfaces to the user.
  - Plug-in Application Binary Interface (ABI): an internal interface designed for developers to easily write modules for a specific platform with ease.

- Bundled plug-ins
  - IPMI: Both a basic IPMI 1.5 platform support and, just recently, the HPI B.01.01 AdvancedTCA Mapping Specification. This plug-in is based on the popular OpenIPMI library for low level IPMI access and supports rack-mount servers as well as AdvancedTCA.
  - IPMIDirect: This plug-in communicates directly with the Shelf Manager in an AdvancedTCA system over RMCP. It is mainly developed for AdvancedTCA systems.
  - SNMP for BladeCenter: Supports BladeCenter data based on SNMP.
  - SYSFS: Nonintelligent devices such as ADM sensor chips or lm sensors that are exposed via the Linux operating system's SYSFS properties.

- HPI Remoting
  Open HPI also includes an answer to a gap in the HPI specification, which does not address how to transact HPI over remote connections. The specification left it to be implementation specific and Open HPI provides support for lightweight HPI remoting based on client/server concepts. The Core library is used as a client library as well for Management Consoles to access the HPI API. However, in the remoting scenario, Open HPI provides an alternative client library with marshalling combined with an Open HPI daemon running on the managed system to de-marshal the requests coming over the wire.

- HPI View
  - A GTK based HPI GUI application.

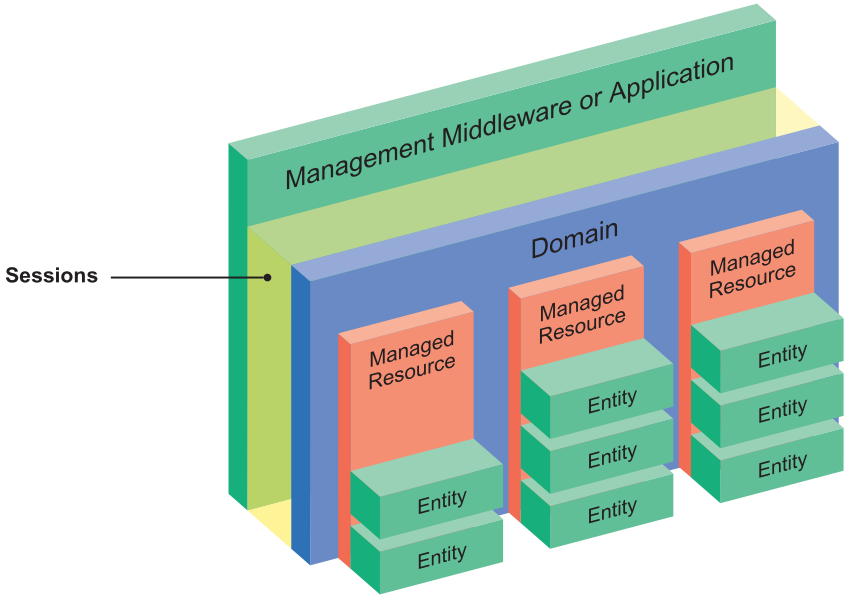

Figure 1

HPI-to-AdvancedTCA mapping specification

The SA Forum provided an additional specification in December 2005 addressing the details of support for AdvancedTCA systems using HPI.

AdvancedTCA is a PICMG standard specifying a bladed chassis with an extensive built-in management capability based on IPMI. In order to address the complexities of a bladed architecture and the expanded functionality, PICMG had to create extensions to IPMI specific for AdvancedTCA to achieve the desired goals beyond those that IPMI 1.5 and 2.0 provided.

Previously, I had highlighted the legacy of HPI in IPMI. However, HPI B.01.01 did not fully expose all AdvancedTCA features and capabilities without further clarification and expansion of the HPI data-types and API. Therefore the HPI-to-AdvancedTCA mapping specification was needed.

Features such as failover, active and standby shelf managers, Advanced Mezzanine Cards, and dual/redundant shelf Field Replaceable Unit (FRU) devices needed to be expanded on or added altogether to HPI in order for implementations to be able to map the AdvancedTCA exposed data using HPI.

The Open HPI Project has been actively developing support for the mapping specification since its announcement, and the work is publicly available in the project's source control (CVS) on SourceForge.

In order to support the HPI B.01.01 and the new mapping specification, Open HPI had to work extensively with the OpenIPMI project. OpenIPMI provides the Linux Kernel IPMI Device Driver in addition to a C Binary Interface to low-level IPMI commands via a shared library.

Conclusion

HPI is becoming a greatly recognized and required interface in the overall telecom and enterprise platform management area. Many Telecom Equipment Manufacturers (TEMs) are requiring that their platform vendors provide integrated support or at least validate HPI (and in most cases Open HPI) support of the platform.

With the introduction of the HPI-to-AdvancedTCA Mapping Specification, the SA Forum strengthened its position as a leader in standardizing the TEM and overall telecom interfaces, thus moving the industry forward and away from proprietary, antiquated, and costly solutions.

The benefits HPI offers its audience are tangible and have been proven over time. Will we see a time when HPI is the de facto interface into a system management? I cannot answer that. However, I predict that HPI will keep growing as market needs increase with the faster and escalating adoption of AdvancedTCA. In addition, vendors are looking for open standards-based solutions and more open source software, and hardware, while platform makers are focusing more on overall platform management.

Tariq Shureih *is staff software engineer in the Open Source Technology Center at Intel Corporation. He has worked at Intel since 1999, and has been the maintainer of the Open HPI project for the past two years.*

To learn more, contact Tariq at tariq.shureih@intel.com.

- Open Source Technology Intel Corporation
  www.intel.com
- Service Availability Forum
  www.saforum.org
- Open HPI
  www.openhpi.org