Automated Generation of FRU Devices Inventory Records for xTCA Devices

Jan Wychowaniak, Dariusz Makowski, *Member, IEEE*, Piotr Perek, and Andrzej Napieralski, *Senior Member, IEEE*

***Abstract*—The Advanced Telecommunications Computing Architecture (ATCA) and Micro Telecommunications Computing Architecture (µTCA) standards, intended for high-performance applications, offer an array of features that are compelling from the industry use perspective, like high reliability (99,999%) or hot-swap support.**

The standards incorporate the Intelligent Platform Management Interface (IPMI) for the purpose o advanced diagnostics and operation control. This standard imposes support for non-volatile Field Replaceable Unit (FRU) information for specific components of an ATCA/µTCA-based system, which would typically include description of a given component. The Electronic Keying (EK) mechanism is capable of using this information for ensuring more reliable cooperation of the components.

The FRU Information for the ATCA/µTCA implementation elements may be of sophisticated structure. This paper focuses on a software tool facilitating the process of assembling this information, the goal of which is to make it more effective and less error-prone.

***Index Terms*—Intelligent Platform Management Interface, Advanced Telecommunications Computing Architecture, Micro Telecommunications Computing Architecture, Hot swap, Advanced Mezzanine Card, Electronic Keying, Field Replaceable Unit, PCI Industrial Computer Manufacturers Group**

I. Introduction

The Advanced Telecommunications Computing Architecture (ATCA) and Micro Telecommunications Computing Architecture (MicroTCA or µTCA, both collectively referred to as xTCA) are a complementary pair of standards by the PCI Industrial Computer Manufacturers Group (PICMG) for facilitation of high-performance computing. The ATCA aims for very high capacity applications, while µTCA is intended to be used for cost sensitive systems offering potentially lower capacity [1].

Both standards offer a number of features that are important from the implementation point of view, like high reliability or hot-swap support. Also, both are modular and their implementations exist as ensembles of components. Most essential of them are Advanced Mezzanine Cards (AMC, in ATCA and µTCA) and Boards (ATCA). These elements pay the functionality needed for the implemented systems to be useful [1, 2]. The standards implement Intelligent Platform Management Interface (IPMI) as a means of advanced operation control and diagnostics. The IPMI infrastructure describes support for storing and accessing non-volatile Field Replaceable Unit (FRU) information for specific components of a xTCA-based system. The FRU Information is used to provide information about the boards that the unit holding it is placed on. Such data storage typically includes description of every major system component. It contains items such as model, serial number, part number and more [3].

All major FRUs are expected to have some kind of FRU Information device (e.g. EEPROM) from which the named information pieces can be read by using software [3].

The FRU Information is also expected to be reachable on occurrence of system failure or abnormal operation conditions, when standard FRU access mechanisms, based on the use of a main system processor, become out of service [4].

The FRU information in xTCA-based systems should also store data required to set up and enable connections for exchanging data between the system components, which would be governed by the Electronic Keying (EK) mechanism. The EK is capable of verifying fabric compatibility between the components prior to enabling the connection, which significantly reduces the risk of harming the hardware [5]. The FRU information data is taken advantage of for this process [1, 2].

The activity of assembling the FRU information required for desired parts of the xTCA-based system by-hand may be time-consuming and error-prone. Therefore there has been developed a software tool, the purpose of which is to boost this process and increase its efficiency, as well as to reduce the probability of error occurrence. This is achieved by providing significant automation to generation of the FRU records.

The ATCA standard was formerly intended to be used as a base for the Low Level Radio Frequency (LLRF) system for the X-Ray Free Electron Laser (X-FEL) [6]. Now it is the µTCA implementation that is under development as a new base, but significant amount of effort invested in working out ATCA-based solutions results in broad knowledge base and

The authors are scholarship holders of project entitled "Innovative education ..." supported by European Social Fund. The research leading to these results has received funding from the European Commission under the FP7 Research Infrastructures project EuCARD, grant agreement no. 227579.

J. Wychowaniak, D. Makowski, P. Perek, A. Napieralski are with the Technical University of Lodz, Department of Microelectronics and Computer Science 93-590 Lodz, Poland (e-mail: janwych@dmcs.pl).

experience, together with substantial amount of prototype hardware. Both of those should contribute to further projects that may be brought into existence on the ATCA-shaped basis. The automated FRU Information generation means presented in this paper can be also applied to systems based on the μTCA architecture, which in total creates potentially broad domain of applicability for this solution.

II. The FRU Information

A. The Purpose

The FRU Information is present at every major FRU component of a system that supports the IPMI functionality. It is used basically to provide a definition of hardware description information regarding the component (e.g. an ATCA Board, AMC module) that the FRU Information storage device is located on [2, 4]. It is required that the FRU Information of a given FRU can reveal, at a minimum, the part number or version number, which can be read through platform management software.

Additionally, more information can be stored, as with accordance to the ATCA-related specifications [2, 7]. It may concern e.g.:

- power supplies parameters
- power management, both for the level of an entire ATCA Shelf, as well as for Carrier Boards
- cooling management
- Ethernet interface and IP addressing configuration for a Shelf Manager
- current requirements for AMC modules
- description of the point-to-point connections for data transport, as implemented on:
  - the Shelf Backplane
  - Boards (connections to the Base, Fabric, and Update Channel Interfaces), including Carriers
  - AMC modules

B. The Structure

The intelligent platform management FRU Information storage format, as to the extent of the paper interest, is defined across several specifications. The base structure and fundamental rules of its composition are laid out at the IPMI level [4]. Other ATCA-oriented specifications that depend on the IPMI, like the PICMG 3.0 (ATCA) or PICMG AMC.0 (for AMC modules) perform further refinements of the definition in order to tailor it for the use with ATCA/AMC-related system components [2, 7]. Fig. 1 draws the structure of FRU Information with specifying its elements, which a particular management controller may take advantage of, subject to the kind of a FRU device it governs. The illustration is aimed at containing some exemplary set of those elements in order to convey their basic organisation and relation scheme. It also indicates the specification a particular FRU Info element is defined by.

The IPMI standard provides definitions of some most fundamental FRU Information content sections, also referred to as areas. The Common Header area, for instance, the only mandatory area among all possible, serves as a table of contents for the entire dataset, describing what other areas are

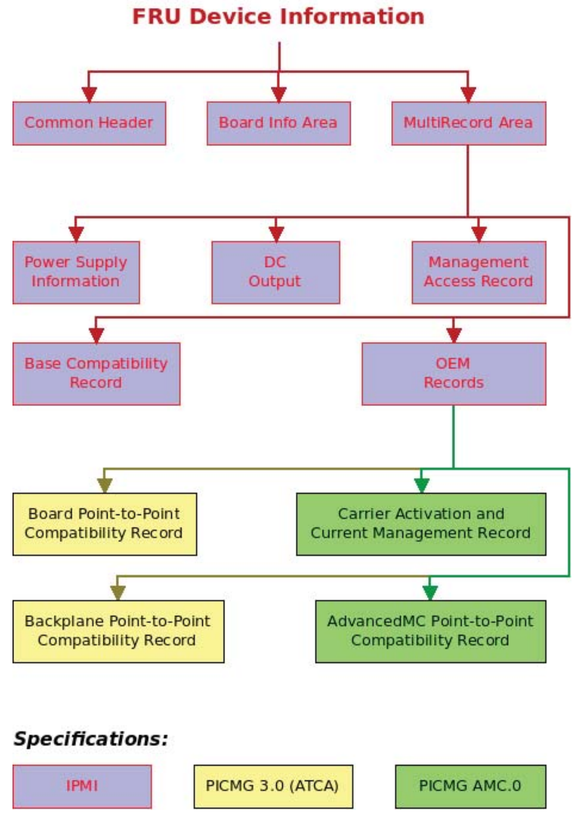


Fig. 1. The FRU Information structure diagram.

present and where they can be found. The Board and the Product Info areas provide serial and part numbers, among other information, about the board that the FRU Information device is located on. A distinctive part is the MultiRecord area, a region split up further into sub-regions, which are referred to as records. This area, provided with a basic set of predefined records, allows for defining additional, custom records for parties that would like to extend the types and the amount of data that FRU Information is capable of storing. The PICMG specifications take advantage of that and provide a wider set of records for the architecture-specific use.

C. Relationship with the Electronic Keying

The Electronic Keying (EK) represents a means by which it is possible for the mandatory shelf management infrastructure to dynamically determine whether the interfaces on a system component are compatible with the interconnects governed by its supervising unit [2, 5]. The mechanism operates the same way, whether connections between ATCA Boards via the Backplane, or between AMC modules residing in an ATCA Carrier Board, are considered. As for the interfaces used for setting up the connections, the PICMG 3.0 specification enumerates a group of such links. These are: a single Base Interface – the 10/100/1000 BASE-T, and a group of Fabric interfaces: Ethernet, Infiniband, StarFabric and PCI Express [2]. The Specification also leaves the room for defining proprietary interfaces. An example of those may be the interfaces introduced during the development of the ATCA Carrier Board hardware intended previously for the use in the LLRF. They fall into two kinds: Fabric, for which there are the Rocket IO interface and the Low Latency Link interface, and Update Channel, represented by the Rocket IO Update interface [5].

Fig. 2 gives a summary on the interfaces and channels relation. The foundation for the E-Keying process is the E-Keying entries present as FRU Information in every major FRU component of the Shelf, including all Boards. Those entries describe the Base Interface, Fabric Interface, and Update Channel Interface implemented by those FRUs.

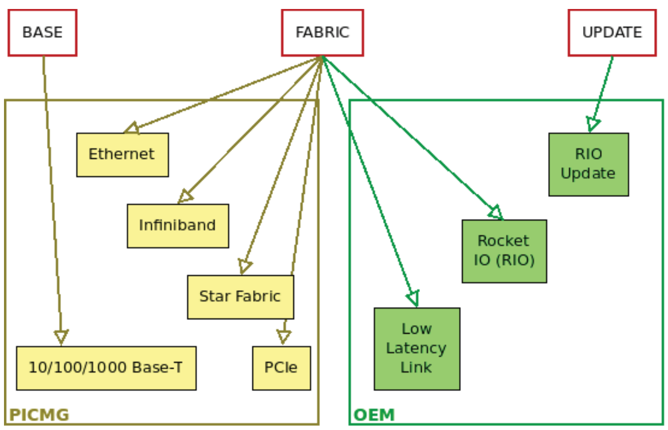


Fig. 2. The interfaces and channels relation.

D. Current Implementation

The FRU Information can be stored in a non-volatile memory, e.g. an EEPROM. In such case the system management software needs to know the particular details of how to access the storage device, so only a limited number of storage device types are supported for the purpose of direct access.

Another, and in fact preferable, approach is to use a management controller to provide the interface to the FRU Information Device. This method enables for additional data integrity tests and error handling, which is not available at a direct access to a non-volatile storage device.

A sort of the latter approach has been applied to the process of development of the prototype ATCA-based system, which was previously being considered for the use in the X-FEL LLRF control system. The FRU Information to be associated with the ATCA Carrier Board management controller was implemented as a block of C language code, with the intention of it being included directly in the controller firmware code. The FRU Information code has a form of a “char”-type array, an exemplary snippet of which can be seen in Fig. 3. The figure depicts a fragment of an implementation of the MultiRecord area, namely the Board Point-to-Point Connectivity record with an exemplary content, as following the format defined by the AMC specification [7]. Such structure is made a part of the management controller code.

```
/* MRA Record: Board Point-to-Point Connectivity */
/* ---- header ----*/
0xC0,           /* Record Type = OEM */
0x02,           /* End of list + Record Format */
50,             /* Data length of Record */
0,              /* Record Checksum */
0,              /* Header Checksum */
/* ----- data -----*/
0x5A,           /* Manufacturer ID */
0x31,           /* Manufacturer ID */
0x00,           /* Manufacturer ID */
0x14,           /* PCIMG Record ID (14h) - Board Point-to-Point */
0x00,           /* Record Format Version (00h) */
0x02,           /* OEM GUID Count */
'L','o','w',' ','L','a','t','e','n','c','y',' ','L','i','n','k',
'R','o','c','k','e','t',' ','I','O',' ',' ',' ',' ',' ',' ',' ',
/* Link Descriptor: */
0b00000001,     /* [7:6] Interface: Base, [5:0] Channel nr: 1 */
0b00010110,     /* [7:4] Type LSB: 1000 Base-T, [3:0] Ports: 1,2 */
0b00000000,     /* [7:4] Type Ext, [3:0] Type MSB: 1000 Base-T */
0x00,           /* Link Grouping ID - Channel Independent */
/* Link Descriptor: */
0b01000001,     /* [7:6] Interface: Fabric, [5:0] Channel nr: 1 */
0b00001000,     /* [7:4] Type LSB: OEM LLL, [3:0] Ports: 3 */
0b00001111,     /* [7:4] Type Ext, [3:0] Type MSB: OEM LLL */
0x00,           /* Link Grouping ID - Channel Independent */
/* Link Descriptor: */
0b01000001,     /* [7:6] Interface: Fabric, [5:0] Channel nr: 1 */
0b00011100,     /* [7:4] Type LSB: OEM RIO, [3:0] Ports: 2,3 */
0b00001111,     /* [7:4] Type Ext, [3:0] Type MSB: OEM RIO */
0x00,           /* Link Grouping ID - Channel Independent */
```

Fig. 3. A fragment of exemplary FRU Information implementation.

E. Motivation

The composition of the content sections that the FRU Information consists of (areas or records), as defined by the specifications, is considerably complicated, provided a management controller programmer intends to assemble it by hand:

- The sections need to include information of their length,
- The Common Header area needs to be provided with offsets to other, laid out consecutively, sections,
- Elements of the content to be later interpreted as text strings need to be provided with a piece of data containing their length,
- The MultiRecord area also includes offsets and some records position relations information as a part of its organisation,
- The areas and records are required to be provided with checksums, the values of which need to be calculated basing of their content,
- Some areas are expected to have the length equal a multiple of 8 bytes, therefore an appropriate number of unused bytes to be filled with the padding value is included,
- Finally, much of the information the programmer is expected to provide needs to be converted to a form that would be capable of being stored as single bytes, multi-byte groups or even packaged into fractions of bytes (a single or multiple bits).

All those values need to be calculated prior to putting it into the structure. This, together with a care of the general data organisation, burdens the programmer with a serious amount of additional effort to be made. It also creates a vast amount of opportunities for making an error that could be later difficult to track down.

What is more, being able to compose the FRU Information content in accordance to the specifications, in which its form is defined, makes the programmer expected to be familiar to a great detail with the selected areas of the specifications.

The FRU Information automated generation means that is discussed here is aimed at relieving the programmer from the need of making all the calculations. Also, it contains knowledge of structure of the FRU Information content, thus reducing the necessity to delve into its definition details.

The C array output format is considered convenient enough to be maintained for the purposes of further tests with the architecture. Therefore, the result to be produced by the FRU Information automation means is planned to reflect this approach and to reproduce this appearance. The FRU Information content generated in such a way is a piece of code that is syntactically valid, as from the point of view of the C language.

III. The FRU Info Generator

The discussed automated FRU Information generation means take form of a Graphical User Interface (GUI)-based software entity. Fig. 4 depicts the idea of the user interface of the application.

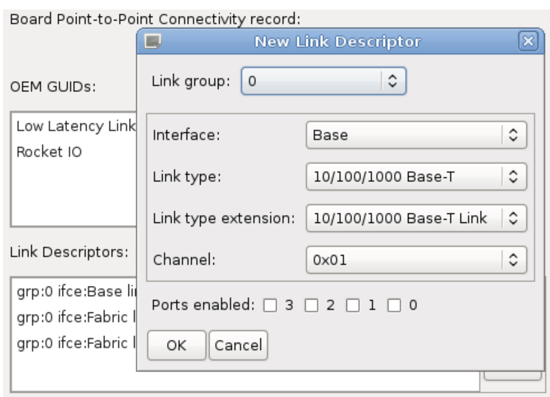


Fig. 4. An exemplary fragment of the application GUI.

By using the application instead of a text editor or even some more advanced kind of code development application, the programmer provides data to be placed into the FRU Information by interacting with simple graphical widgets, which together compose intuitive interface. For every FRU Information content section that the application supports, the application provides visual panel that enables the programmer to provide all the data for the particular section that is required by a corresponding specification.

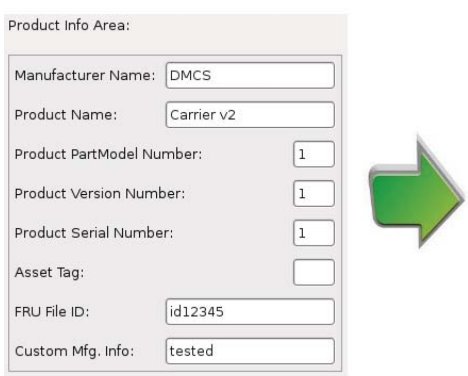
Fig. 5 illustrates how a particular set of exemplary data provided to the interface of the application corresponds to a result that is produced basing on that data. Here the Product Info area FRU Information section is being used as a showcase of the application operation. The part to the right shows a fragment of C code that the application generates from the data set provided into the visual components seen to the left.

The generated code features all the assisting data like areas and text strings lengths, offsets, padding fields, checksums etc. All this data is automatically calculated by the application upon generation of the output text, which is properly formatted and syntactically valid C code, ready to be included in the management controller firmware.

IV. Internal Structure of the Application

The development process of the application included tests performed with two of the xTCA-based hardware components: an µTCA-based universal controller development board and an ATCA Carrier test board, which was originally designed to contribute to the X-FEL LLRF control system before the technology choice change. The application has proved its applicability and usefulness for providing the FRU Information content, since it could be directly used in the code of the controllers.

The application itself is created with the Java programming language on the top of the Eclipse Rich Client Platform (RCP) [9]. The factors supporting this choice were the suitability for creating graphical user interfaces combined with high level of software portability, which made it easy to produce versions of the application for multiple platforms, like Linux or MS Windows.




```
/* Area Name: Product Info */
0x01,           /* Product Area Format Version */
0x06,           /* Product Info Area Length */
0x19,           /* Language Code: English */
0xC4,           /* Manufacturer Name length */
'D', 'M', 'C', 'S',
0xCA,           /* Product Name length */
'C', 'a', 'r', 'r', 'i', 'e', 'r', ' ', 'v', '2',
0xC2,           /* Product Part/Model Number length */
'0',            /* Product Part/Model Number */
'1',            /* Product Part/Model Number */
0xC2,           /* Product Version Number length */
'0',            /* Product Version Number */
'1',            /* Product Version Number */
0xC2,           /* Product Serial Number length */
'0',            /* Product Serial Number */
'1',            /* Product Serial Number */
0xC2,           /* Asset Tag length */
'0',            /* Asset Tag */
' ',            /* Asset Tag */
0xC7,           /* FRU File ID length */
'i', 'd', '1', '2', '3', '4', '5',
0xC6,
't', 'e', 's', 't', 'e', 'd',
0xC1,           /* no-more-info-fields indicator */
0x00,           /* Checksum */
/* -48- */
```

Fig. 5. Data provided to the application (left) and the corresponding output (right)

Fig. 6 gives an overview of the internal structure of the application.

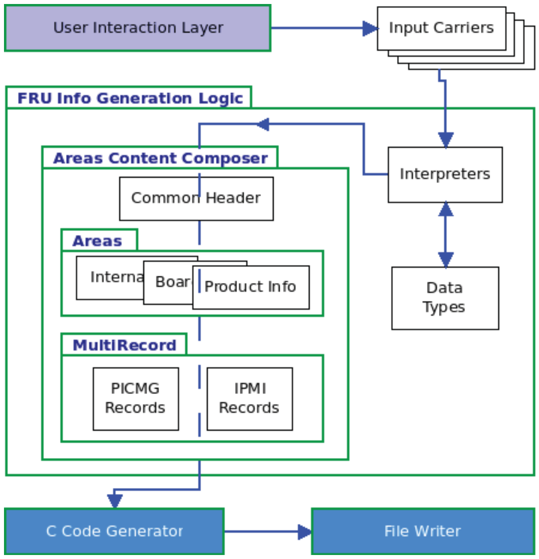


Fig. 6. The block diagram of the application structure.

Data acquired from the user by the means of the application GUI are to be interpreted by the generation logic. The latter implements the definitions of the FRU Information organisation, in the way it is laid out by the specifications. Therefore data obtained from the user can be put into logical structures that represent all the areas that the application supports. Those structures are then passed to the C code generation block, the task of which is to convert the data into text form respecting the C language syntax. This text is the final output and the application stores it in a text file, from where it can be e.g. freely copied and used elsewhere. For instance, in a code editor, where a management controller code is developed.

V. Conclusion and Plans for Further Development

The intelligent platform management is one of the features that lay upon the foundation of the exceptionally high reliability and system availability that the xTCA-based systems are characterised by.

A crucial feature of the intelligent platform management is its inventory management capability. FRU Information, that is present at every major xTCA system component, accompanies the management controller in the task of managing system features like power, cooling, and interconnect operation of intelligent devices.

Using the discussed application for the purpose of generation of the Platform Management FRU Information storage contents offers a series of benefits to this process. The software accepts input by means that are friendly and intuitive to its user, and then fully cares of the shape of generated data and its validity. This way the amount of time and effort spent on the development of the management infrastructure of an intelligent device may be reduced to a noticeable extent. What is more, the risk of making a calculation error within the FRU Information content is eradicated, thus limiting potential sources of the component malfunction. The care of compliance of the generated result to the architecture specifications, which is taken by the application, is a factor that brings additional reduction to the degree of need for the programmer's focus.

Further development plans for the application assume interest in the Sensor Data Records, which describe sensors and other related objects within the xTCA shelf management system. Support for automated generation of their content is planned to be incorporated in the application, thus making it a more complete aid for an intelligent platform management infrastructure implementer. The application may also be made available via the Internet, for wider use, to potentially interested parties.

References

[1] MicroTCA PICMG MTCA.0 Shortform, Revision 1.0, “MicroTCA Short Form Specification”, 26 September 2006
[2] AdvancedTCA PICMG 3.0 Revision 2.0, “AdvancedTCA Base Specification”, 28 October 2005
[3] Intel Hewlett-Packard NEC DELL, “Intelligent Platform Management Specification version 1.5”, Document Revision 1.1, 20 February 2002
[4] IPMI Platform Management FRU Information Storage Definition v1.0, Revision 1.1, September 27, 1999
[5] Predki, P., Makowski, D., “Hot-plug Based Activation and Deactivation of ATCA FRU Devices” MIXDES 2009
[6] Jezynski, T., Simrock, S., “The Low Level Radio Frequency System Architecture for the European X-FEL” MIXDES 2006
[7] PICMG AMC.0 Revision 1.0, "Advanced Mezzanine Card Base Specification", April 28, 2006
[8] Pigeon Point Systems, “Pigeon Point IPM Sentry Shelf Manager User Guide, ShMM-300/ShMM-500” Release 2.1.1
[9] The Eclipse Foundation, “The Rich Client Platform”, available from http://www.eclipse.org/home/categories/rcp.php

Andrzej Napieralski received the M.Sc. and Ph.D. degrees from the Technical University of Lodz (TUL) in 1973 and 1977, respectively, and a D.Sc. degree in electronics from the Warsaw University of Technology (Poland) and in microelectronics from the Université de Paul Sabatié (France) in 1989. Since 1996 he has been a Director of the Department of Microelectronics and Computer Science. In 2002 he has been elected and in 2005 reelected as the Vice-President of TUL. He is an author or co-author of over 900 publications and editor of 18 conference proceedings and 12 scientific Journals. He supervised 44 PhD theses; four of them received the prize of the Prime Minister of Poland. In 2008 he received the Degree of Honorary Doctor of Yaroslaw the Wise Novgorod State University (Russia).

Jan Wychowaniak graduated from the Technical University of Lodz in 2010, earning MSc in Electronics and Telecommunications, in the field of microprocessors systems and reprogrammable devices. He is a PhD student at TUL, with the main fields of interests in software development and digital systems. He is involved in development of diagnostics and management facilities for systems based on the ATCA and uTCA architectures.

Dariusz Makowski received PhD degree in electrical engineering at the Department of Microelectronics and Computer Science Technical University of Lodz in 2006. His main areas of interests are digital electronics, embedded systems and programmable devices. He is engaged in the development of xTCA standards for High Energy Physics. He is involved in the design of distributed data acquisition and control systems based on ATCA, MTCA and AMC standards.

Piotr Perek received the Master of Science degree in the field of Electronics and Telecommunications from Technical University of Lodz in 2010. He continues his education as a PhD student at the Department of Microelectronics and Computer Science TUL. His interests include digital electronics and software development both for computers as well as for embedded systems. His recent research has involved the development of controllers for IPMI management layer for control systems based on ATCA and AMC standards.