Profile Error: This profile contains 1 errors (search for 'Error:')

Document Number: XMP1011

Date: 2011-08-31

Version: 1.0.3m

Example Physical Asset Profile

IMPORTANT: This specification is not a standard. It does not necessarily reflect the views of the DMTF or all of its members. Because this document is a Work in Progress, this specification may still change, perhaps profoundly. This document is available for public review and comment until the stated expiration date.

This document expires on: **2012-02-28**.

Target version for DMTF Standard: **1.0.3**.

Provide any comments through the DMTF Feedback Portal: http://www.dmtf.org/standards/feedback

Document Type: Specification

Document Status: Work in Progress

Document Language: en-US

Copyright notice

CONTENTS

21

Figures

22

Tables

23

Foreword

This document was prepared by the Physical Platform Profiles Working Group and Server Management Working Group.

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability.

Design Note: This document contains design notes (like this one), that provide information about the way the document is written, or to demonstrate certain things. Such design notes would not appear in a released version of this document.

Design Note: This MRP document represents DSP1011 1.0.2 as a machine readable profile in MRP 1.1 format. Since machine readable profiles need to be compliant to DSP1001 1.1, this document utilizes the newly introduced concepts, such as adaptations, features and collaboration diagrams.

Acknowledgements

DMTF acknowledges the following individuals for their contributions to this document:

- Jon Hass, Dell Inc. (editor)
- Khachatur Papanyan, Dell Inc. (editor)
- Jeff Hilland, HP (editor)
- Hemal Shah, Broadcom Corporation (editor)
- Christina Shaw, HP
- Aaron Merkin, IBM
- Jeff Lynch, IBM
- Arvind Kumar, Intel
- Perry Vincent, Intel
- John Leung, Intel

Document conventions

Any text in this document is in normal text font, with the following exceptions:

- References to clause names use normal text font; if they consist of more than one word, the clause name is quoted using double quotes, such as in "CIM elements".
- Important terms that are used for the first time are marked in *italics* .
- The usage of terms link to the term definition defined in the "Terms and definitions" clause, enabling easy navigation to the term definition.

Example Physical Asset Profile

45

1 Scope

46 The Physical Asset profile extends the management capability of the referencing profiles by adding the capability to represent the physical aspects of elements in the managed environment, the relationship with their logical aspects, and the implementation of this profile.

47

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated or versioned references, only the edition cited (including any corrigenda or DMTF update versions) applies. For undated and unversioned references, the latest published edition of the referenced document (including any corrigenda or DMTF update versions) applies.

48 DMTF DSP0004, *CIM Infrastructure Specification 2.5*, http://www.dmtf.org/standards/published_documents/DSP0004_2.5.pdf

49 DMTF DSP0223, *Generic Operations 1.0*, http://www.dmtf.org/standards/published_documents/DSP0223_1.0.pdf

50 DMTF DSP1001, *Management Profile Specification Usage Guide 1.1*, http://www.dmtf.org/standards/published_documents/DSP1001_1.1.pdf

51 DMTF XMP1033, *Example Profile Registration Profile (sample profile in DSP2023) 1.0*, http://www.dmtf.org/standards/published_documents/DSP2023_1.0.zip

52 ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*, http://isotc.iso.org/livelink/livelink?func=ll&objId=4230456&objAction=browse&sort=subtype

53

3 Terms and definitions

In this document, some terms have a specific meaning beyond the normal English meaning. Those terms are defined in this clause.

54

3.1 General

55 The terms "shall" ("required"), "shall not", "should" ("recommended"), "should not" ("not recommended"), "may", "need not" ("not required"), "can" and "cannot" in this document are to be interpreted as described in ISO/IEC Directives, Part2 , Annex H. The terms in parenthesis are alternatives for the preceding term, for use in exceptional cases when the preceding term cannot be used for linguistic reasons. Note that ISO/IEC Directives, Part2 , Annex H specifies additional alternatives. Occurrences of such additional alternatives shall be interpreted in their normal English meaning in this document.

56 The terms "clause", "subclause", "paragraph", "annex" in this document are to be interpreted as described in ISO/IEC Directives, Part2 , Clause 5.

57 The terms "normative" and "informative" in this document are to be interpreted as described in ISO/IEC Directives, Part2 , Clause 3. In this document, clauses, subclauses or annexes indicated with "(informative)" as well as notes and examples do not contain normative content.

The terms defined in DSP0004, DSP0223, and DSP1001 apply to this document.

58

The following additional terms are defined in this document.

59 **3.2**

60 **connecting element**

A physical element that has the ability to connect with a physical connector . It is modeled through the abstract base adaptation AbstractConnectingElement .

61 **3.3**

62 **delimited substring**

A substring of a vendor compatibility string that starts at the beginning of the vendor compatibility string and ends at the end of the string, or at a character that precedes a colon (:) within the string. See features PackageSlotCompatibilityForPackage and PackageSlotCompatibilityForSlot .

63 **3.4**

64 **field replaceable unit**

65 A physical element that can be replaced in the field. Also sometimes called a *part* .

66 **3.5**

67 **FRU information**

Information that supports the replacement of a FRU in the field.

68 **3.6**

69 **FRU number**

A string that identifies the FRU . The uniqueness of that string and potential correspondence to a part number is defined by the manufacturer of the FRU . The FRU number is part of the FRU information .

70 **3.7**

71 **physical component**

A physical element that cannot be further decomposed. It is modeled through the adaptation PhysicalComponent .

72 **3.8**

73 **physical element**

An element in the managed environment that has a physical presence (e.g. could be touched). It is modeled through the abstract base adaptation AbstractPhysicalElement .

74 **3.9**

75 **physical package**

A physical element that has the ability to contain (or house, host) other physical elements . It is modeled through the adaptation PhysicalPackage .

76 **3.10**

77 **physical connector**

A physical element that has the ability to connect with a connecting element . It is modeled through the PhysicalConnector adaptation.

78 **3.11**

79 **slot**

A place to plug a physical package into.

80

3.12

81 **system chassis**

The physical package of the managed system.

82

4 Symbols and abbreviated terms

This clause defines the symbols and abbreviations used in this document.

83

4.1 General

84 The abbreviations defined in DSP0004, DSP0223, and DSP1001 apply to this document.

85 The following additional abbreviations are defined in this document.

86 **4.2**

87 **FRU**

Field Replaceable Unit

88

5 Synopsis

89 **Profile name:** Example Physical Asset

90 **Version:** 1.0.3

91 **Organization:** DMTF

92 **Abstract:** No

93 **Profile type:** Component

94 **Schema:** DMTF CIM 2.22 (experimental)

95 **Central class adaptation:** AbstractPhysicalElement

96 **Scoping class adaptation:** AbstractSystem

97 **Scoping path:** Realizes, LogicalDevice, SystemDevice

98 The Physical Asset profile extends the management capability of the referencing profiles by adding the capability to represent the physical aspects of physical elements in the managed environment, the relationship with their logical aspects, and the implementation of this profile. Physical aspects include asset, inventory, and other descriptive physical information. The profile does not cover the geographic location of the physical assets.

99 Table 1 identifies the profile references defined in this profile.

100 **Table 1 – Profile references**

Profile reference name	Profile name	Organi-zation	Version	Relation-ship	Description
PRP	Profile Registration	DMTF	1.0	Mandatory	Used to represent the implementation of this profile.

101 Table 2 identifies the features defined in this profile.

102

Table 2 – Features

Feature	Requirement	Description
FRUInformation	Optional	See 7.1.1.
FieldReplaceability	Optional	See 7.1.2.
LogicalElementRelation	Optional	See 7.1.3.
PackageSlotCompatibilityForPackage	Optional	See 7.1.4.
PackageSlotCompatibilityForSlot	Optional	See 7.1.5.
ManagedSystemAsComputerSystem	ConditionalExclusive	See 7.1.6.
ManagedSystemAsSystem	ConditionalExclusive	See 7.1.7.
SystemChassisForSystem	Optional	See 7.1.8.
SystemChassisForComputerSystem	Optional	See 7.1.9.
ConfigurationCapacity	Optional	See 7.1.10.
PackageContainmentHierarchy	Optional	See 7.1.11.
PackageSlotConnection	Optional	See 7.1.12.
ConnectorConnection	Optional	See 7.1.13.

103 Table 3 identifies the class adaptations defined in this profile.

104 **Table 3 – Adaptations**

Adaptation	Elements	Requirement	Description
Instantiated, embedded and abstract adaptations			
AbstractSystem	CIM_System	See derived adaptations	See 7.2.2.
ComputerSystem	CIM_ComputerSystem	ConditionalExclusive	See 7.2.3.
System	CIM_System	ConditionalExclusive	See 7.2.4.
ComputerSystemPackage	CIM_ComputerSystemPackage	ConditionalExclusive	See 7.2.5.
SystemPackaging	CIM_SystemPackaging	ConditionalExclusive	See 7.2.6.
SystemDevice	CIM_SystemDevice	Optional	See 7.2.7.
LogicalDevice	CIM_LogicalDevice	Conditional	See 7.2.8.
Realizes	CIM_Realizes	Conditional	See 7.2.9.
AbstractPhysicalElement	CIM_PhysicalElement	See derived adaptations	See 7.2.10.
AbstractConnectingElement	CIM_PhysicalElement	See derived adaptations	See 7.2.11.
AbstractPhysicalConnector	CIM_PhysicalConnector	See derived adaptations	See 7.2.12.
AbstractPhysicalComponent	CIM_PhysicalComponent	See derived adaptations	See 7.2.13.
AbstractPhysicalPackage	CIM_PhysicalPackage	See derived adaptations	See 7.2.14.
PhysicalPackage	CIM_PhysicalPackage	Mandatory	See 7.2.15.
PhysicalFrame	CIM_PhysicalFrame	Optional	See 7.2.16.
Chassis	CIM_Chassis	Optional	See 7.2.17.
SystemChassis	CIM_Chassis	Optional	See 7.2.18.
Rack	CIM_Rack	Optional	See 7.2.19.
Card	CIM_Card	Optional	See 7.2.20.
PhysicalConnector	CIM_PhysicalConnector	Mandatory	See 7.2.21.

Adaptation	Elements	Requirement	Description
Slot	CIM_Slot	Optional	See 7.2.22.
PhysicalComponent	CIM_PhysicalComponent	Mandatory	See 7.2.23.
Chip	CIM_Chip	Optional	See 7.2.24.
PhysicalMemory	CIM_PhysicalMemory	Optional	See 7.2.25.
PhysicalAssetCapabilities	CIM_PhysicalAssetCapabilities	Conditional	See 7.2.26.
ElementCapabilities	CIM_ElementCapabilities	Conditional	See 7.2.27.
ConfigurationCapacity	CIM_ConfigurationCapacity	Conditional	See 7.2.28.
ElementCapacityForPackage	CIM_ElementCapacity	Conditional	See 7.2.29.
ElementCapacityForSystemChassis	CIM_ElementCapacity	Conditional	See 7.2.30.
Container	CIM_Container	Mandatory	See 7.2.31.
ElementInConnector	CIM_ElementInConnector	Optional	See 7.2.32.
ConnectedTo	CIM_ConnectedTo	Optional	See 7.2.33.
Indications and exceptions			
This profile does not define any such adaptations.			

105 Table 4 identifies the use cases and state descriptions defined in this profile.

106 **Table 4 – Use cases and state descriptions**

Name	Description
State description: SystemChassis	See 8.1.
State description: FanPackage	See 8.2.
Use case: FindScopingInstance1	See 8.3.
Use case: FindScopingInstance2	See 8.4.
State description: PhysicalTopology	See 8.5.
State description: PhysicalMemory	See 8.6.
State description: ConfigurationCapacity1	See 8.7.
State description: ConfigurationCapacity2	See 8.8.
State description: NetworkPortConnector	See 8.9.
Use case: DeterminePartNumberOfComponent	See 8.10.
Use case: GetPhysicalInventoryOfSystem	See 8.11.
Use case: GetPhysicalInventoryOfSystemChassis	See 8.12.
Use case: DetermineSlotIsEmpty	See 8.13.
Use case: GetFanCapacityOfChassis	See 8.14.
Use case: GetMaxFanCapacityOfChassis	See 8.15.

107

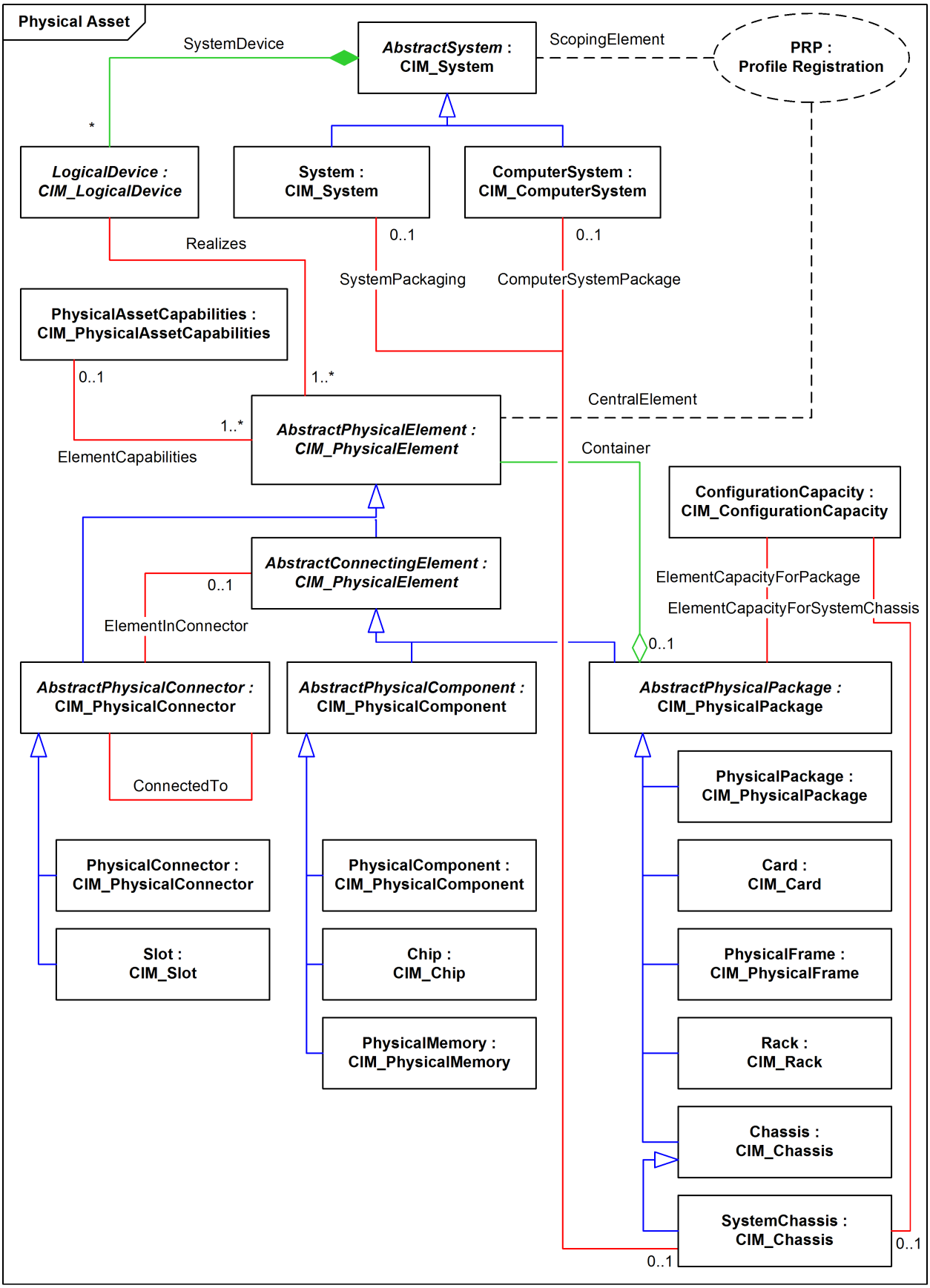
6 Description

108 Design Note: This description clause demonstrates how to define subclause headings in the profile, and how they line up with those headings that are created by the DSP8029 XSLT stylesheet.

109

6.1 General

110


Physical Asset
SystemDevice
*
AbstractSystem : CIM_System
ScopingElement
PRP : Profile Registration
LogicalDevice : CIM_LogicalDevice
System : CIM_System
ComputerSystem : CIM_ComputerSystem
Realizes
0..1
0..1
SystemPackaging
ComputerSystemPackage
PhysicalAssetCapabilities : CIM_PhysicalAssetCapabilities
0..1
1..*
1..*
ElementCapabilities
AbstractPhysicalElement : CIM_PhysicalElement
CentralElement
Container
ConfigurationCapacity : CIM_ConfigurationCapacity
AbstractConnectingElement : CIM_PhysicalElement
0..1
ElementInConnector
ElementCapacityForPackage
ElementCapacityForSystemChassis
0..1
AbstractPhysicalConnector : CIM_PhysicalConnector
AbstractPhysicalComponent : CIM_PhysicalComponent
AbstractPhysicalPackage : CIM_PhysicalPackage
ConnectedTo
PhysicalPackage : CIM_PhysicalPackage
PhysicalConnector : CIM_PhysicalConnector
PhysicalComponent : CIM_PhysicalComponent
Card : CIM_Card
Slot : CIM_Slot
Chip : CIM_Chip
PhysicalFrame : CIM_PhysicalFrame
PhysicalMemory : CIM_PhysicalMemory
Rack : CIM_Rack
Chassis : CIM_Chassis
SystemChassis : CIM_Chassis
0..1
0..1

111

Figure 1 – DMTF collaboration structure diagram

112

Todo: This subclause so far is a plain copy of the profile text and needs to be updated.

113 A Physical Element (see section 3.15) describes the physical properties, including the FRU information, of a managed element. The capabilities of the Physical Elements are described by the properties of the CIM_PhysicalAssetCapabilities class. The Physical Elements could be associated to the logical representation of the managed element through the CIM_Realizes association. The enclosures or chassis of the managed systems are represented by a CIM_PhysicalElement or CIM_Chassis instance that is associated to the CIM_System/CIM_ComputerSystem instance through the CIM_SystemPackaging/ CIM_ComputerSystemPackage association and are referred to as a System Chassis (see section 3.17). Configuration capacity of the System Chassis is also represented within this profile by CIM_ConfigurationCapacity instances.

114 Physical Elements can be also arranged in a topology. The CIM_Container, CIM_ConnectedTo, and CIM_ElementInConnector associations are used to associate the Physical Elements and create the physical topology of the managed elements.

115 Figure 1 also represents the ecosystem of Physical Asset Profile classes, illustrating their relationship with classes of referencing profiles. The referencing profiles can identify the subclass of CIM_PhysicalElement to be used for representing the physical aspects of the managed element. For example, the referencing profiles that contain a CIM_LogicalDevice subclass can restrict the associated subclass of CIM_PhysicalPackage to CIM_PhysicalMemory for instantiation of the Physical Asset Profile. Such restrictions will be described in the referencing profiles.

116 The Physical Asset Profile is advertised through the CIM_RegisteredProfile instance.

117 The Physical Asset Profile can be instantiated to represent a combination of the following scenarios:

118 • the physical aspects of a managed system, such as the FRU information for the chassis (see section 7.6)

119 • the physical aspects of a specific managed element, such as the FRU information of a fan (see section 7.3)

120 • the physical hierarchy of a managed system, such as the relationship between chassis, slots, and packages (see section 7.8)

121 • the configuration capacity of a managed element, such as the minimum and maximum number of certain types of packages that the managed system can handle (see section 7.7)

122

6.2 Relationship to logical elements

123 Physical elements represented by AbstractPhysicalElement instances can expose the relationship with the logical elements of which they represent a physical aspect and that are represented by LogicalDevice instances, if the LogicalElementRelation feature is implemented for the physical element . For details, see LogicalElementRelation .

124

6.3 FRU information of physical elements

125 Physical elements represented by AbstractPhysicalElement instances can expose their FRU information if the FRUInformation feature is implemented for the physical element . For details, see FRUInformation .

126

6.4 Plug compatibility of physical packages

127 Physical packages represented by PhysicalPackage instances and slots represented by Slot instances can expose their plug compatibility if the PackageSlotCompatibilityForPackage feature is implemented for the physical package and the PackageSlotCompatibilityForSlot feature is implemented for the slot .

128

6.5 System chassis

129 Systems represented by System or ComputerSystem instances can expose their system chassis if the SystemChassisForSystem or the SystemChassisForComputerSystem feature is implemented for the system.

130

6.6 Configuration capacity

131 Physical packages represented by PhysicalPackage instances (including the system itself) can expose their configuration capacity if the ConfigurationCapacity feature is implemented for the physical package . For details, see ConfigurationCapacity .

132

6.7 Physical topology

133 The following aspects of physical topology can be exposed:

134 • Package containment hierarchy: The containment hierarchy of physical packages can be exposed if the PackageContainmentHierarchy feature is implemented. For details, see PackageContainmentHierarchy .

135 • Package to slot connections: The connections between physical packages and slots can be exposed if the PackageSlotConnection feature is implemented. For details, see PackageSlotConnection .

136 • Connections between connectors: The connections between physical connectors can be exposed if the ConnectorConnection feature is implemented. For details, see ConnectorConnection .

137

7 Implementation

138

7.1 Features

139

7.1.1 Feature: FRUInformation

140 **Requirement level:** Optional

The implementation of this feature is not required for a physical element to expose its replaceability in the field (see FieldReplaceability feature), or to be replaceable.

142 Implementing this feature provides support for exposing FRU information for a physical element that is represented by an AbstractPhysicalElement instance.

143 FRU information is represented by a combination of the values of the following properties of the AbstractPhysicalElement instance:

144 • Manufacturer

145 • Model

146 • PartNumber

147

- SerialNumber
- SKU

If the SKU property is non-NULL, it shall be used to convey the FRU number .

This feature can be made available to clients at the granularity of AbstractPhysicalElement instances.

It can be concluded that the feature is available for a AbstractPhysicalElement instance if:

- The following OCL derivation constraint evaluates to a Boolean value of True.

  OCL context: A AbstractPhysicalElement instance.

  ```
  derive: self.ElementCapabilities::Capabilities.FRUInfoSupported
  ```

Otherwise, it can be concluded that the feature is not available.

7.1.2 Feature: FieldReplaceability

Requirement level: Optional

The implementation of this feature or replaceability in the field is not required for a physical element to expose its FRU information (see FRUInformation feature).

Implementing this feature provides support for exposing the ability for physical elements to be replaced in the field, via values TRUE or FALSE of the CanBeFRUed property of the AbstractPhysicalElement instance representing the physical element .

This feature can be made available to clients at the granularity of AbstractPhysicalElement instances.

It can be concluded that the feature is available for a AbstractPhysicalElement instance if:

- The following OCL derivation constraint evaluates to a Boolean value of True.

  OCL context: A AbstractPhysicalElement instance.

  ```
  derive: self.CanBeFRUed.isNotNull()
  ```

Otherwise, it can be concluded that the feature is not available.

7.1.3 Feature: LogicalElementRelation

Requirement level: Optional

This feature provides support for exposing the relationship of physical elements to the logical elements of which they represent a physical aspect.

Design Note: The profile is not particularly clear as to whether physical and logical elements are two distinct managed object types, or different aspects of the same managed object type (and if so, what is that one managed object type). This MRP profile did not attempt to resolve that questions and lives with any ambiguities resulting from that.

This feature can be made available to clients at the granularity of AbstractPhysicalElement instances.

It can be concluded that the feature is available for a AbstractPhysicalElement instance if:

- The following OCL derivation constraint evaluates to a Boolean value of True.

  OCL context: A AbstractPhysicalElement instance.

  ```
  derive: self.Realizes::Dependent->size() > 0
  ```

Otherwise, it can be concluded that the feature is not available.

7.1.4 Feature: PackageSlotCompatibilityForPackage

Requirement level: Optional

Implementing this feature for a physical package represented by an PhysicalPackage instance and for a slot represented by a Slot instance provides support for indicating the compatibility of that physical package with that slot . Only if a physical package is compatible with a slot , it can be plugged into that slot .

Note that physical components represented by PhysicalComponent instances may be compatible with physical connectors represented by PhysicalConnector instances without being able to indicate that compatibility.

See the AbstractPhysicalPackage . VendorCompatibilityStrings and Slot . VendorCompatibilityStrings properties for a description of their format.

This feature can be made available to clients at the granularity of AbstractPhysicalPackage instances.

It can be concluded that the feature is available for a AbstractPhysicalPackage instance if:

- The following OCL derivation constraint evaluates to a Boolean value of True.

  OCL context: A AbstractPhysicalPackage instance.

  ```
  derive: self.VendorCompatibilityStrings.isNotNull()
  ```

Otherwise, it can be concluded that the feature is not available.

7.1.5 Feature: PackageSlotCompatibilityForSlot

Requirement level: Optional

Implementing this feature for a physical package represented by an PhysicalPackage instance and for a slot represented by a Slot instance provides support for indicating the compatibility of that physical package with that slot . Only if a physical package is compatible with a slot , it can be plugged into that slot .

Note that physical components represented by PhysicalComponent instances may be compatible with physical connectors represented by PhysicalConnector instances without being able to indicate that compatibility.

See the AbstractPhysicalPackage . VendorCompatibilityStrings and Slot . VendorCompatibilityStrings properties for a description of their format.

This feature can be made available to clients at the granularity of Slot instances.

It can be concluded that the feature is available for a Slot instance if:

- The following OCL derivation constraint evaluates to a Boolean value of True.

  OCL context: A Slot instance.

  ```
  derive: self.VendorCompatibilityStrings.isNotNull()
  ```

Otherwise, it can be concluded that the feature is not available.

7.1.6 Feature: ManagedSystemAsComputerSystem

Requirement level: Conditional exclusive

Condition:

The following is NOT true:

- The ManagedSystemAsSystem feature is implemented.

Implementing this feature for the profile will represent the managed system using the ComputerSystem adaptation.

This feature can be made available to clients at the granularity of an implementation of this profile.

It can be concluded that the feature is available if:

- The following OCL derivation constraint evaluates to a Boolean value of True.

  OCL context: A RegisteredProfile instance for this profile.

```
derive:
self.CIM_Realizes::Dependent->CIM_SystemDevice::GroupComponent->isTypeOf(CIM_ComputerSystem)
```

Otherwise, it can be concluded that the feature is not available.

7.1.7 Feature: ManagedSystemAsSystem

Requirement level: Conditional exclusive

Condition:

The following is NOT true:

- The ManagedSystemAsComputerSystem feature is implemented.

Implementing this feature for the profile will represent the managed system using the System adaptation.

This feature can be made available to clients at the granularity of an implementation of this profile.

It can be concluded that the feature is available if:

- The following OCL derivation constraint evaluates to a Boolean value of True.

  OCL context: A RegisteredProfile instance for this profile.

```
derive:
self.CIM_Realizes::Dependent->CIM_SystemDevice::GroupComponent->isTypeOf(CIM_ComputerSystem)
== False
```

Otherwise, it can be concluded that the feature is not available.

7.1.8 Feature: SystemChassisForSystem

Requirement level: Optional

Implementing this feature for a system provides support for exposing the system chassis of the system, represented by a Chassis instance.

This feature can be made available to clients at the granularity of System instances.

It can be concluded that the feature is available for a System instance if:

- The following OCL derivation constraint evaluates to a Boolean value of True.

  OCL context: A System instance.

```
derive: self.SystemPackaging::Dependent->size() > 0
```

Otherwise, it can be concluded that the feature is not available.

7.1.9 Feature: SystemChassisForComputerSystem

Requirement level: Optional

Implementing this feature for a system provides support for exposing the system chassis of the system, represented by a Chassis instance.

This feature can be made available to clients at the granularity of ComputerSystem instances.

It can be concluded that the feature is available for a ComputerSystem instance if:

- The following OCL derivation constraint evaluates to a Boolean value of True.

  OCL context: A ComputerSystem instance.

  ```
  derive: self.ComputerSystemPackage::Dependent->size() > 0
  ```

Otherwise, it can be concluded that the feature is not available.

7.1.10 Feature: ConfigurationCapacity

Requirement level: Optional

The implementation of this feature for a physical package provides support for exposing the configuration capacity of that physical package , including of a system chassis .

This feature can be made available to clients at the granularity of AbstractPhysicalPackage instances.

It can be concluded that the feature is available for a AbstractPhysicalPackage instance if:

- The following OCL derivation constraint evaluates to a Boolean value of True.

  OCL context: A AbstractPhysicalPackage instance.

  ```
  derive: self.ElementCapacityForPackage::Capacity->size() > 0
  ```

Otherwise, it can be concluded that the feature is not available.

7.1.11 Feature: PackageContainmentHierarchy

Requirement level: Optional

This feature provides support for exposing the containment hierarchy of physical packages as follows:

- For any physical element that resides within a physical package , that relationship shall be represented by a Container instance.

This feature can be made available to clients at the granularity of AbstractPhysicalElement instances.

It can be concluded that the feature is available for a AbstractPhysicalElement instance if:

- The following OCL derivation constraint evaluates to a Boolean value of True.

  OCL context: A AbstractPhysicalElement instance.

  ```
  derive: self.Container::PartComponent->size() > 0
  ```

Otherwise, it can be concluded that the feature is not available.

7.1.12 Feature: PackageSlotConnection

Requirement level: Optional

This feature provides support for exposing the connections between physical packages and slots as follows:

- For any physical package that is is plugged in or connected to a slot , that relationship shall be represented by an ElementInConnector instance.

This feature can be made available to clients at the granularity of AbstractPhysicalPackage instances.

It can be concluded that the feature is available for a AbstractPhysicalPackage instance if:

- The following OCL derivation constraint evaluates to a Boolean value of True.

  OCL context: A AbstractPhysicalPackage instance.

  ```
  derive: self.ElementInConnector::Antecedent->size() > 0
  ```

Otherwise, it can be concluded that the feature is not available.

7.1.13 Feature: ConnectorConnection

Requirement level: Optional

Physical connectors may be connected to each other at a certain level of abstraction. For example, the ports of a non-switching network router may be considered connected at the abstraction level of electrical connections, while the ports of an IP-layer switch may be considered connected at the abstraction level of IP communication. The level of abstraction and whether each connector provides the exact same capabilities as one that is connected to it, is not modeled in this profile.

This feature provides support for exposing connections between physical connectors as follows:

- For any physical connector that is connected to another one, that relationship shall be represented by a ConnectedTo instance.

This feature can be made available to clients at the granularity of AbstractPhysicalConnector instances.

It can be concluded that the feature is available for a AbstractPhysicalConnector instance if:

- The following OCL derivation constraint evaluates to a Boolean value of True.

  OCL context: A AbstractPhysicalConnector instance.

  ```
  derive: self.ConnectedTo::Dependent->size() > 0
  ```

Otherwise, it can be concluded that the feature is not available.

7.2 Adaptations

7.2.1 Conventions

This profile defines operation requirements based on DSP0223.

For adaptations of ordinary classes and of associations, the requirements for operations are defined in adaptation-specific subclauses of subclause 7.2.

For association traversal operation requirements that are specified only in the elements table of an adaptation (i.e., without operation-specific subclauses), the names of the association adaptations to be traversed are listed in the elements table.

The default initialization requirement level for property requirements is optional.

The default modification requirement level for property requirements is optional.

This profile repeats the effective values of certain Boolean qualifiers as part of property, method parameter, or method return value requirements. The following convention is established: If the name of a qualifier is listed, its effective value is True; if the qualifier name is not listed, its effective value is False. The convention is applied in the following cases:

- In: indicates that the parameter is an input parameter
- Out: indicates that the parameter is an output parameter
- Key: indicates that the property is a key (that is, its value is part of the instance path)
- Required: indicates that the element value shall be non-Null
- Null OK: indicates explicitly that the element value may be Null for mandatory, conditional or conditional exclusive properties. This information is not specified as a qualifier in the schema but as an indicator in the profile.

7.2.2 Adaptation: AbstractSystem: CIM_System

This abstract adaptation provides a basis for derived adaptations representing specific types of systems.

Adaptation type: Ordinary class

Implementation type: Abstract

Requirement level: Defined by its derived adaptations

7.2.3 Adaptation: ComputerSystem: CIM_ComputerSystem

This adaptation models managed systems.

Implementation of this adaptation is mutually exclusive with the System adaptation; the choice is determined by referencing profiles.

Adaptation type: Ordinary class

Implementation type: Instantiated

Requirement level: Conditional exclusive

Condition:

The ManagedSystemAsComputerSystem feature is implemented.

Table 5 – ComputerSystem: Element requirements

Element	Requirement	Description
Base adaptations		
AbstractSystem	Optional	See AbstractSystem.
Operations		
Associators( )	Mandatory	
AssociatorNames( )	Mandatory	
References( )	Mandatory	
ReferenceNames( )	Mandatory	

7.2.4 Adaptation: System: CIM_System

This adaptation models managed systems.

Implementation of this adaptation is mutually exclusive with the ComputerSystem adaptation; the choice is determined by referencing profiles.

Adaptation type: Ordinary class

Implementation type: Instantiated

A concrete subclass of the abstract schema class CIM_System needs to be implemented.

Requirement level: Conditional exclusive

Condition:

The ManagedSystemAsSystem feature is implemented.

Table 6 – System: Element requirements

Element	Requirement	Description
Base adaptations		
AbstractSystem	Optional	See AbstractSystem.
Operations		
Associators( )	Mandatory	
AssociatorNames( )	Mandatory	
References( )	Mandatory	
ReferenceNames( )	Mandatory	

7.2.5 Adaptation: ComputerSystemPackage: CIM_ComputerSystemPackage

7.2.5.1 General

Adaptation type: Association class

Implementation type: Instantiated

Requirement level: Conditional exclusive

Condition:

The ManagedSystemAsComputerSystem feature is implemented.

This adaptation models the relationship between managed systems modeled with ComputerSystem and the system chassis of these managed systems modeled with SystemChassis .

Table 7 – ComputerSystemPackage: Element requirements

Element	Requirement	Description
Properties		
Antecedent	Mandatory	Key, see 7.2.5.2
Dependent	Mandatory	Key, see 7.2.5.3
PlatformGUID	Mandatory	See 7.2.5.4
Operations		
GetInstance( )	Mandatory	

7.2.5.2 Property: Antecedent

Requirement level: Mandatory

Reference kind: REF-typed

Constraints:

- Referenced instances shall be of class adaptation SystemChassis.
- The multiplicity of this association end is 1 .. 1

7.2.5.3 Property: Dependent

Requirement level: Mandatory

Reference kind: REF-typed

Constraints:

- Referenced instances shall be of class adaptation ComputerSystem.
- The multiplicity of this association end is 0 .. 1

7.2.5.4 Property: PlatformGUID

Requirement level: Mandatory

Constraint:

OCL constraint in the context of a ComputerSystemPackage instance:

```
inv: self.PlatformGUID.mrpMatchesPattern('^[0-9A-F]{32}$|0')
```

7.2.6 Adaptation: SystemPackaging: CIM_SystemPackaging

7.2.6.1 General

Adaptation type: Association class

Implementation type: Instantiated

Requirement level: Conditional exclusive

Condition:

The ManagedSystemAsComputerSystem feature is implemented.

This adaptation models the relationship between managed systems modeled with System and the system chassis of these managed systems modeled with SystemChassis .

Table 8 – SystemPackaging: Element requirements

Element	Requirement	Description
Properties		
Antecedent	Mandatory	Key, see 7.2.6.2
Dependent	Mandatory	Key, see 7.2.6.3
Operations		
GetInstance( )	Mandatory	

7.2.6.2 Property: Antecedent

Requirement level: Mandatory

Reference kind: REF-typed

Constraints:

- Referenced instances shall be of class adaptation SystemChassis.
- The multiplicity of this association end is 1 .. 1

7.2.6.3 Property: Dependent

Requirement level: Mandatory

Reference kind: REF-typed

Constraints:

- Referenced instances shall be of class adaptation System.
- The multiplicity of this association end is 0 .. 1

7.2.7 Adaptation: SystemDevice: CIM_SystemDevice

7.2.7.1 General

Adaptation type: Association class

Implementation type: Instantiated

Requirement level: Optional

This adaptation models the relationship between managed systems and the logical elements on these systems.

Table 9 – SystemDevice: Element requirements

Element	Requirement	Description
Properties		
GroupComponent	Mandatory	Key, see 7.2.7.2
PartComponent	Mandatory	Key, see 7.2.7.3

7.2.7.2 Property: GroupComponent

Requirement level: Mandatory

Reference kind: REF-typed

Constraints:

- Referenced instances shall be of class adaptation AbstractSystem.
- The multiplicity of this association end is 1 .. 1

7.2.7.3 Property: PartComponent

Requirement level: Mandatory

Reference kind: REF-typed

Constraints:

- Referenced instances shall be of class adaptation LogicalDevice.
- The multiplicity of this association end is 0 .. *

7.2.8 Adaptation: LogicalDevice: CIM_LogicalDevice

This adaptation models logical elements that are realized by a physical element .

Adaptation type: Ordinary class

Implementation type: Instantiated

A concrete subclass of the abstract schema class CIM_LogicalDevice needs to be implemented.

Requirement level: Conditional

Condition:

The LogicalElementRelation feature is implemented.

Table 10 – LogicalDevice: Element requirements

Element	Requirement	Description
Operations		
Associators( )	Mandatory	
AssociatorNames( )	Mandatory	
References( )	Mandatory	
ReferenceNames( )	Mandatory	

7.2.9 Adaptation: Realizes: CIM_Realizes

7.2.9.1 General

Adaptation type: Association class

Implementation type: Instantiated

Requirement level: Conditional

Condition:

The LogicalElementRelation feature is implemented.

This adaptation models the relationship between physical elements and the logical elements realized by them.

Table 11 – Realizes: Element requirements

Element	Requirement	Description
Properties		
Antecedent	Mandatory	Key, see 7.2.9.2
Dependent	Mandatory	Key, see 7.2.9.3

7.2.9.2 Property: Antecedent

Requirement level: Mandatory

Reference kind: REF-typed

Constraints:

- Referenced instances shall be of class adaptation AbstractPhysicalElement.
- The multiplicity of this association end is 0 .. *

7.2.9.3 Property: Dependent

Requirement level: Mandatory

Reference kind: REF-typed

Constraints:

- Referenced instances shall be of class adaptation LogicalDevice.
- The multiplicity of this association end is 0 .. *

7.2.10 Adaptation: AbstractPhysicalElement: CIM_PhysicalElement

7.2.10.1 General

Adaptation type: Ordinary class

Implementation type: Abstract

Requirement level: Defined by its derived adaptations

This abstract adaptation provides a basis for derived adaptations by modeling properties and behaviours of physical elements that are common for more specific types of physical elements and provides a common end point for referencing associations.

Table 12 – AbstractPhysicalElement: Element requirements

Element	Requirement	Description
Properties		
Manufacturer	Conditional	See 7.2.10.2
Model	Conditional	See 7.2.10.3
SerialNumber	Conditional	See 7.2.10.4
PartNumber	Conditional	See 7.2.10.5
SKU	Conditional	See 7.2.10.6
CanBeFRUed	Conditional	See 7.2.10.7
ElementName	Mandatory	See 7.2.10.8
Operations		
Associators( )	Mandatory	
AssociatorNames( )	Mandatory	
References( )	Mandatory	
ReferenceNames( )	Mandatory	

7.2.10.2 Property: Manufacturer

Requirement level: Conditional

Condition:

The FRUInformation feature is implemented.

Constraint:

OCL constraint in the context of a AbstractPhysicalElement instance:

```
inv: self.Manufacturer.mrpMatchesPattern('[^WSP]+')
```

Explanation:

Design Note: Clarify whether this pattern means "no whitespace".

7.2.10.3 Property: Model

Requirement level: Conditional

Condition:

The FRUInformation feature is implemented.

Constraint:

OCL constraint in the context of a AbstractPhysicalElement instance:

```
inv: self.Model.mrpMatchesPattern('[^WSP]+')
```

Explanation:

Design Note: Clarify whether this pattern means "no whitespace".

7.2.10.4 Property: SerialNumber

Requirement level: Conditional

Condition:

The FRUInformation feature is implemented.

Constraint:

OCL constraint in the context of a AbstractPhysicalElement instance:

```
inv: self.SerialNumber.mrpMatchesPattern('[^WSP]+')
```

Explanation:

Design Note: Clarify whether this pattern means "no whitespace".

7.2.10.5 Property: PartNumber

Requirement level: Conditional

Condition:

The FRUInformation feature is implemented.

Constraint:

OCL constraint in the context of a AbstractPhysicalElement instance:

```
inv: self.PartNumber.mrpMatchesPattern('[^WSP]+')
```

Explanation:

Design Note: Clarify whether this pattern means "no whitespace".

7.2.10.6 Property: SKU

Requirement level: Conditional

Condition:

The FRUInformation feature is implemented.

Constraint:

OCL constraint in the context of a AbstractPhysicalElement instance:

```
inv: self.SKU.mrpMatchesPattern('[^WSP]+')
```

Explanation:

Design Note: Clarify whether this pattern means "no whitespace".

7.2.10.7 Property: CanBeFRUed

Requirement level: Conditional

Condition:

The FieldReplaceability feature is implemented.

7.2.10.8 Property: ElementName

Requirement level: Mandatory

Constraint:

OCL constraint in the context of a AbstractPhysicalElement instance:

```
inv: self.ElementName.isNotNull()
```

7.2.11 Adaptation: AbstractConnectingElement: CIM_PhysicalElement

This abstract adaptation provides a basis for derived adaptations by modeling properties and behaviours of connecting elements that are common for more specific types of connecting elements and provides a common end point for referencing associations.

Adaptation type: Ordinary class

Implementation type: Abstract

Requirement level: Defined by its derived adaptations

Table 13 – AbstractConnectingElement: Element requirements

Element	Requirement	Description
Base adaptations		
AbstractPhysicalElement	Optional	See AbstractPhysicalElement.
Operations		
Associators( )	Mandatory	
AssociatorNames( )	Mandatory	
References( )	Mandatory	
ReferenceNames( )	Mandatory	

7.2.12 Adaptation: AbstractPhysicalConnector: CIM_PhysicalConnector

This abstract adaptation provides a basis for derived adaptations by modeling properties and behaviours of physical connectors that are common for more specific types of connectors and provides a common end point for referencing associations.

Adaptation type: Ordinary class

Implementation type: Abstract

487 **Requirement level:** Defined by its derived adaptations

488 **Table 14 – AbstractPhysicalConnector: Element requirements**

Element	Requirement	Description
Base adaptations		
AbstractPhysicalComponent	Optional	See AbstractPhysicalComponent.
Properties		
CreationClassName	Mandatory	Key
Tag	Mandatory	Key
ConnectorLayout	Mandatory	
Operations		
GetInstance( )	Mandatory	
EnumerateInstances( )	Mandatory	
EnumerateInstanceNames( )	Mandatory	

489

7.2.13 Adaptation: AbstractPhysicalComponent: CIM_PhysicalComponent

This abstract adaptation provides a basis for derived adaptations by modeling properties and behaviours of physical components (i.e. physical elements that cannot be further decomposed) that are common for more specific types of components and provides a common end point for referencing associations.

490 **Adaptation type:** Ordinary class

491 **Implementation type:** Abstract

492 **Requirement level:** Defined by its derived adaptations

493 **Table 15 – AbstractPhysicalComponent: Element requirements**

Element	Requirement	Description
Base adaptations		
AbstractConnectingElement	Optional	See AbstractConnectingElement.
Properties		
CreationClassName	Mandatory	Key
Tag	Mandatory	Key
Operations		
GetInstance( )	Mandatory	
EnumerateInstances( )	Mandatory	
EnumerateInstanceNames( )	Mandatory	

494

7.2.14 Adaptation: AbstractPhysicalPackage: CIM_PhysicalPackage

495

7.2.14.1 General

496 **Adaptation type:** Ordinary class

497 **Implementation type:** Abstract

498 **Requirement level:** Defined by its derived adaptations

This abstract adaptation provides a basis for derived adaptations by modeling properties and behaviours of physical packages (i.e. physical elements that have the ability to contain other physical elements) that are common for more specific types of packages and provides a common end point for referencing associations.

Table 16 – AbstractPhysicalPackage: Element requirements

Element	Requirement	Description
Base adaptations		
AbstractConnectingElement	Optional	See AbstractConnectingElement.
Properties		
CreationClassName	Mandatory	Key
Tag	Mandatory	Key
PackageType	Mandatory	
VendorCompatibilityStrings	Mandatory	See 7.2.14.2
Operations		
GetInstance( )	Mandatory	
EnumerateInstances( )	Mandatory	
EnumerateInstanceNames( )	Mandatory	
Associators( )	Mandatory	
AssociatorNames( )	Mandatory	
References( )	Mandatory	
ReferenceNames( )	Mandatory	

7.2.14.2 Property: VendorCompatibilityStrings

Requirement level: Mandatory

See feature PackageSlotCompatibilityForPackage .

Constraint:

OCL constraint in the context of a AbstractPhysicalPackage instance:

```
inv: self.VendorCompatibilityStrings.mrpMatchesPattern('[^:]+(:[^:]+)+')
```

7.2.15 Adaptation: PhysicalPackage: CIM_PhysicalPackage

This adaptation models physical packages for which no other sibling adaptation in this profile matches.

Adaptation type: Ordinary class

Implementation type: Instantiated

Requirement level: Mandatory

Table 17 – PhysicalPackage: Element requirements

Element	Requirement	Description
Base adaptations		
AbstractPhysicalPackage	Optional	See AbstractPhysicalPackage.

7.2.16 Adaptation: PhysicalFrame: CIM_PhysicalFrame

This adaptation models physical packages that are frames.

Adaptation type: Ordinary class

Implementation type: Instantiated

Requirement level: Optional

Table 18 – PhysicalFrame: Element requirements

Element	Requirement	Description
Base adaptations		
AbstractPhysicalPackage	Optional	See AbstractPhysicalPackage.
Operations		
GetInstance( )	Mandatory	
EnumerateInstances( )	Mandatory	
EnumerateInstanceNames( )	Mandatory	

7.2.17 Adaptation: Chassis: CIM_Chassis

7.2.17.1 General

Adaptation type: Ordinary class

Implementation type: Instantiated

Requirement level: Optional

This adaptation models physical packages that are chassis'.

Table 19 – Chassis: Element requirements

Element	Requirement	Description
Base adaptations		
AbstractPhysicalPackage	Optional	See AbstractPhysicalPackage.
Properties		
PackageType	Mandatory	See 7.2.17.2
ChassisPackageType	Mandatory	
Operations		
GetInstance( )	Mandatory	
EnumerateInstances( )	Mandatory	
EnumerateInstanceNames( )	Mandatory	

7.2.17.2 Property: PackageType

Requirement level: Mandatory

Constraint:

OCL constraint in the context of a Chassis instance:

```
inv: self.PackageType = 3 /* Chassis/Frame */
```

7.2.18 Adaptation: SystemChassis: CIM_Chassis

This adaptation models physical packages that are the chassis of the managed system (called a system chassis).

Adaptation type: Ordinary class

Implementation type: Instantiated

Requirement level: Optional

Table 20 – SystemChassis: Element requirements

Element	Requirement	Description
Base adaptations		
Chassis	Optional	See Chassis.
Operations		
Associators( )	Mandatory	
AssociatorNames( )	Mandatory	
References( )	Mandatory	
ReferenceNames( )	Mandatory	

7.2.19 Adaptation: Rack: CIM_Rack

7.2.19.1 General

Adaptation type: Ordinary class

Implementation type: Instantiated

Requirement level: Optional

This adaptation models physical packages that are racks.

Table 21 – Rack: Element requirements

Element	Requirement	Description
Base adaptations		
AbstractPhysicalPackage	Optional	See AbstractPhysicalPackage.
Properties		
PackageType	Mandatory	See 7.2.19.2
TypeOfRack	Mandatory	
Operations		
GetInstance( )	Mandatory	
EnumerateInstances( )	Mandatory	
EnumerateInstanceNames( )	Mandatory	

7.2.19.2 Property: PackageType

Requirement level: Mandatory

Constraint:

OCL constraint in the context of a Rack instance:

```
inv: self.PackageType = 2 /* Rack */
```

543

7.2.20 Adaptation: Card: CIM_Card

This adaptation models physical packages that are cards.

544 **Adaptation type:** Ordinary class

545 **Implementation type:** Instantiated

546 **Requirement level:** Optional

547 **Table 22 – Card: Element requirements**

Element	Requirement	Description
Base adaptations		
AbstractPhysicalPackage	Optional	See AbstractPhysicalPackage.
Properties		
HostingBoard	Mandatory	
Operations		
GetInstance( )	Mandatory	
EnumerateInstances( )	Mandatory	
EnumerateInstanceNames( )	Mandatory	

548

7.2.21 Adaptation: PhysicalConnector: CIM_PhysicalConnector

This adaptation models physical connectors for which no other sibling adaptation in this profile matches.

549 **Adaptation type:** Ordinary class

550 **Implementation type:** Instantiated

551 **Requirement level:** Mandatory

552 **Table 23 – PhysicalConnector: Element requirements**

Element	Requirement	Description
Base adaptations		
AbstractPhysicalConnector	Optional	See AbstractPhysicalConnector.

553

7.2.22 Adaptation: Slot: CIM_Slot

554

7.2.22.1 General

555 **Adaptation type:** Ordinary class

556 **Implementation type:** Instantiated

557 **Requirement level:** Optional

This adaptation models physical packages that are slots.
Slots have the optional ability to expose their plug compatibility, for details see feature PackageSlotCompatibilityForSlot .

558

Table 24 – Slot: Element requirements

Element	Requirement	Description
Base adaptations		
AbstractPhysicalConnector	Optional	See AbstractPhysicalConnector.
Properties		
ConnectorLayout	Mandatory	
VendorCompatibilityStrings	Mandatory	See 7.2.22.2
Operations		
GetInstance( )	Mandatory	
EnumerateInstances( )	Mandatory	
EnumerateInstanceNames( )	Mandatory	

7.2.22.2 Property: VendorCompatibilityStrings

Requirement level: Mandatory

See feature PackageSlotCompatibilityForSlot .

Constraint:

OCL constraint in the context of a Slot instance:

```
inv: self.VendorCompatibilityStrings.mrpMatchesPattern('[^:]+(:[^:]+)+')
```

7.2.23 Adaptation: PhysicalComponent: CIM_PhysicalComponent

This adaptation models physical components for which no other sibling adaptation in this profile matches.

Adaptation type: Ordinary class

Implementation type: Instantiated

Requirement level: Mandatory

Table 25 – PhysicalComponent: Element requirements

Element	Requirement	Description
Base adaptations		
AbstractPhysicalComponent	Optional	See AbstractPhysicalComponent.

7.2.24 Adaptation: Chip: CIM_Chip

This adaptation models physical components that are chips.

Adaptation type: Ordinary class

Implementation type: Instantiated

Requirement level: Optional

Table 26 – Chip: Element requirements

Element	Requirement	Description
Base adaptations		
AbstractPhysicalComponent	Optional	See AbstractPhysicalComponent.

Element	Requirement	Description
Operations		
GetInstance( )	Mandatory	
EnumerateInstances( )	Mandatory	
EnumerateInstanceNames( )	Mandatory	

7.2.25 Adaptation: PhysicalMemory: CIM_PhysicalMemory

7.2.25.1 General

Adaptation type: Ordinary class

Implementation type: Instantiated

Requirement level: Optional

This adaptation models physical components that are memory chips.

Table 27 – PhysicalMemory: Element requirements

Element	Requirement	Description
Base adaptations		
AbstractPhysicalComponent	Optional	See AbstractPhysicalComponent.
Properties		
FormFactor	Mandatory	
Speed	Mandatory	See 7.2.25.2
Capacity	Mandatory	
BankLabel	Mandatory	
Operations		
GetInstance( )	Mandatory	
EnumerateInstances( )	Mandatory	
EnumerateInstanceNames( )	Mandatory	

7.2.25.2 Property: Speed

Requirement level: Mandatory

This property shall represent the speed of the memory chip in a unit of nanoseconds. If the speed of the memory chip is less than one nanosecond or unknown, this property shall have a value of 0. If the speed of the memory chip is variable over time, this property shall have a value of 2^32-1.

7.2.26 Adaptation: PhysicalAssetCapabilities: CIM_PhysicalAssetCapabilities

7.2.26.1 General

Adaptation type: Ordinary class

Implementation type: Instantiated

Requirement level: Conditional

Condition:

The FRUInformation feature is implemented.

590 This adaptation models the capability of physical elements to indicate whether they expose FRU data.

591 **Table 28 – PhysicalAssetCapabilities: Element requirements**

Element	Requirement	Description
Properties		
InstanceID	Mandatory	Key
ElementName	Mandatory	Required
FRUInfoSupported	Mandatory	See 7.2.26.2
Operations		
GetInstance( )	Mandatory	
EnumerateInstances( )	Mandatory	
EnumerateInstanceNames( )	Mandatory	
Associators( )	Mandatory	
AssociatorNames( )	Mandatory	
References( )	Mandatory	
ReferenceNames( )	Mandatory	

592

7.2.26.2 Property: FRUInfoSupported

593 **Requirement level:** Mandatory

594 **Constraint:**

595 OCL constraint in the context of a PhysicalAssetCapabilities instance:

596
```
inv: if self.ElementCapabilities::Element.mrpIsFeatureSupported('FRUInformation')
then self.FRUInfoSupported = true
else self.FRUInfoSupported = false
```

597 Explanation:

If the FRUInformation feature is implemented for an AbstractPhysicalElement instance, the value of this property in the associated instance of this adaptation shall be True. Otherwise, the value of this property shall be False.

598

7.2.27 Adaptation: ElementCapabilities: CIM_ElementCapabilities

599

7.2.27.1 General

600 **Adaptation type:** Association class

601 **Implementation type:** Instantiated

602 **Requirement level:** Conditional

603 **Condition:**

The FRUInformation feature is implemented.

605 Todo: Shall associate an instance of a subclass of CIM_PhysicalElement representing a physical element with the CIM_PhysicalAssetCapabilities instance representing the capabilities of that physical element.

606

Table 29 – ElementCapabilities: Element requirements

Element	Requirement	Description
Properties		
ManagedElement	Mandatory	Key, see 7.2.27.2
Capabilities	Mandatory	Key, see 7.2.27.3
Operations		
GetInstance( )	Mandatory	

7.2.27.2 Property: ManagedElement

Requirement level: Mandatory

Reference kind: REF-typed

Constraints:

- Referenced instances shall be of class adaptation AbstractPhysicalElement.
- The multiplicity of this association end is 1 .. *

7.2.27.3 Property: Capabilities

Requirement level: Mandatory

Reference kind: REF-typed

Constraints:

- Referenced instances shall be of class adaptation PhysicalAssetCapabilities.
- The multiplicity of this association end is 0 .. 1

7.2.28 Adaptation: ConfigurationCapacity: CIM_ConfigurationCapacity

7.2.28.1 General

Adaptation type: Ordinary class

Implementation type: Instantiated

Requirement level: Conditional

Condition:

The ConfigurationCapacity feature is implemented.

Todo: Class adaptation CIM_ConfigurationCapacity advertises the possible configurations of a system chassis.

For each unique value of the VendorCompatibilityStrings property in any Slot instances, a ConfigurationCapacity instance shall exist that has one or more array entry values in its VendorCompatibilityStrings array property that are equal to that unique value.

Table 30 – ConfigurationCapacity: Element requirements

Element	Requirement	Description
Properties		
Name	Mandatory	Key

Element	Requirement	Description
ElementName	Mandatory	
ObjectType	Mandatory	Key
OtherTypeDescription	Mandatory	See 7.2.28.2
MinimumCapacity	Optional	See 7.2.28.3
MaximumCapacity	Mandatory	See 7.2.28.4
Increment	Mandatory	See 7.2.28.5
VendorCompatibilityStrings	Mandatory	See 7.2.28.6
Operations		
GetInstance( )	Mandatory	
EnumerateInstances( )	Mandatory	
EnumerateInstanceNames( )	Mandatory	
Associators( )	Mandatory	
AssociatorNames( )	Mandatory	
References( )	Mandatory	
ReferenceNames( )	Mandatory	

629 #### 7.2.28.2 Property: OtherTypeDescription

630 **Requirement level:** Mandatory

631 **Constraint:**

632 OCL constraint in the context of a ConfigurationCapacity instance:

633
```
inv: if self.ObjectType = 0 /* Other */
then self.OtherTypeDescription != null
else self.OtherTypeDescription = null
```

634 Explanation:

635 If the ObjectType property has a value of 0 (Other), this property shall be non-Null. Otherwise, it shall be Null.

636 #### 7.2.28.3 Property: MinimumCapacity

637 **Requirement level:** Optional

This property should be implemented.

638 #### 7.2.28.4 Property: MaximumCapacity

639 **Requirement level:** Mandatory

A value of 0 shall mean that the maximum capacity is unknown.

640 #### 7.2.28.5 Property: Increment

641 **Requirement level:** Mandatory

A value of 0 shall mean that the increment is unknown.

642

7.2.28.6 Property: VendorCompatibilityStrings

Requirement level: Mandatory

See features PackageSlotCompatibilityForPackage and PackageSlotCompatibilityForSlot .

Constraint:

OCL constraint in the context of a ConfigurationCapacity instance:

```
inv: if self.VendorCompatibilityStrings.mrpMatchesPattern('[^:]+(:[^:]+)+')
```

7.2.29 Adaptation: ElementCapacityForPackage: CIM_ElementCapacity

7.2.29.1 General

Adaptation type: Association class

Implementation type: Instantiated

Requirement level: Conditional

Condition:

The ConfigurationCapacity feature is implemented.

Todo: Shall associate an instance of a subclass of CIM_PhysicalPackage representing a physical package within the chassis, including the chassis itself, with the CIM_ConfigurationCapacity instance representing the configuration capacity of that element.

Table 31 – ElementCapacityForPackage: Element requirements

Element	Requirement	Description
Properties		
Element	Mandatory	Key, see 7.2.29.2
Capacity	Mandatory	Key, see 7.2.29.3
Operations		
GetInstance( )	Mandatory	

7.2.29.2 Property: Element

Requirement level: Mandatory

Reference kind: REF-typed

Constraints:

- Referenced instances shall be of class adaptation AbstractPhysicalPackage.
- The multiplicity of this association end is 0 .. ***Profile Error: Reference "Element" defined in association class adaptation "ElementCapacityForPackage" constrains its minimum multiplicity to "0", which is invalid because it is less than the minimum multiplicity "1" defined in schema association class "CIM_ElementCapacity".**

7.2.29.3 Property: Capacity

Requirement level: Mandatory

Reference kind: REF-typed

Constraints:

- Referenced instances shall be of class adaptation ConfigurationCapacity.
- The multiplicity of this association end is 0 .. *

7.2.30 Adaptation: ElementCapacityForSystemChassis: CIM_ElementCapacity

7.2.30.1 General

Adaptation type: Association class

Implementation type: Instantiated

Requirement level: Conditional

Condition:

The ConfigurationCapacity feature is implemented.

Todo: Shall associate an instance of a subclass of CIM_PhysicalPackage representing a physical package within the chassis, including the chassis itself, with the CIM_ConfigurationCapacity instance representing the configuration capacity of that element.

Table 32 – ElementCapacityForSystemChassis: Element requirements

Element	Requirement	Description
Properties		
Element	Mandatory	Key, see 7.2.30.2
Capacity	Mandatory	Key, see 7.2.30.3
Operations		
GetInstance( )	Mandatory	

7.2.30.2 Property: Element

Requirement level: Mandatory

Reference kind: REF-typed

Constraints:

- Referenced instances shall be of class adaptation SystemChassis.
- The multiplicity of this association end is 1 .. 1

7.2.30.3 Property: Capacity

Requirement level: Mandatory

Reference kind: REF-typed

Constraints:

- Referenced instances shall be of class adaptation ConfigurationCapacity.
- The multiplicity of this association end is 0 .. *

7.2.31 Adaptation: Container: CIM_Container

7.2.31.1 General

Adaptation type: Association class

Implementation type: Instantiated

Requirement level: Mandatory

Todo: Shall associate a physical package (represented by an instance of class adaptation CIM_PhysicalPackage) with the physical elements (represented by instances of class adaptation CIM_PhysicalElement) that reside within the package.

Table 33 – Container: Element requirements

Element	Requirement	Description
Properties		
GroupComponent	Mandatory	Key, see 7.2.31.2
PartComponent	Mandatory	Key, see 7.2.31.3
Operations		
GetInstance( )	Mandatory	

7.2.31.2 Property: GroupComponent

Requirement level: Mandatory

Reference kind: REF-typed

Constraints:

- Referenced instances shall be of class adaptation AbstractPhysicalPackage.
- The multiplicity of this association end is 0 .. 1

7.2.31.3 Property: PartComponent

Requirement level: Mandatory

Reference kind: REF-typed

Constraints:

- Referenced instances shall be of class adaptation AbstractPhysicalElement.
- The multiplicity of this association end is 0 .. *

7.2.32 Adaptation: ElementInConnector: CIM_ElementInConnector

7.2.32.1 General

Adaptation type: Association class

Implementation type: Instantiated

Requirement level: Optional

Todo: Shall associate a CIM_PhysicalConnector or CIM_Slot instance (representing a connector or slot), with physical packages (represented by instances of class adaptation CIM_PhysicalPackage).

Table 34 – ElementInConnector: Element requirements

Element	Requirement	Description
Properties		
Antecedent	Mandatory	Key, see 7.2.32.2
Dependent	Mandatory	Key, see 7.2.32.3
Operations		
GetInstance( )	Mandatory	

7.2.32.2 Property: Antecedent

Requirement level: Mandatory

Reference kind: REF-typed

Constraints:

- Referenced instances shall be of class adaptation AbstractPhysicalConnector.
- The multiplicity of this association end is 0 .. *

7.2.32.3 Property: Dependent

Requirement level: Mandatory

Reference kind: REF-typed

Constraints:

- Referenced instances shall be of class adaptation AbstractConnectingElement.
- The multiplicity of this association end is 0 .. 1

7.2.33 Adaptation: ConnectedTo: CIM_ConnectedTo

7.2.33.1 General

Adaptation type: Association class

Implementation type: Instantiated

Requirement level: Optional

Todo: Shall associate the CIM_PhysicalConnector or CIM_Slot instances that represent connectors that are connected together.

Table 35 – ConnectedTo: Element requirements

Element	Requirement	Description
Properties		
Antecedent	Mandatory	Key, see 7.2.33.2
Dependent	Mandatory	Key, see 7.2.33.3
Operations		
GetInstance( )	Mandatory	

7.2.33.2 Property: Antecedent

Requirement level: Mandatory

Reference kind: REF-typed

Constraints:

- Referenced instances shall be of class adaptation AbstractPhysicalConnector.
- The multiplicity of this association end is 0 .. *

7.2.33.3 Property: Dependent

Requirement level: Mandatory

Reference kind: REF-typed

Constraints:

- Referenced instances shall be of class adaptation AbstractPhysicalConnector.
- The multiplicity of this association end is 0 .. *

8 Use cases and state descriptions

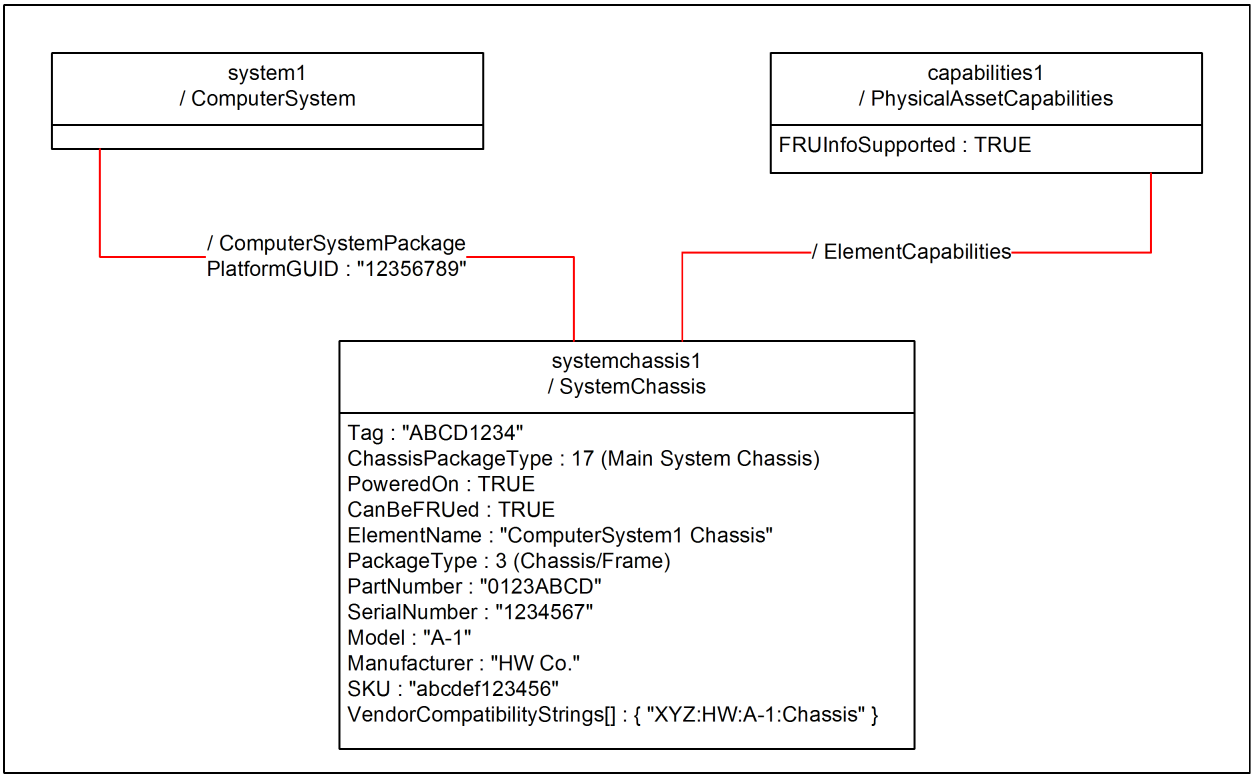
8.1 State description: SystemChassis

The following figure shows a simple object diagram that represents a scenario that conforms to the requirements of this profile. In that scenario, there is a system chassis that is represented by a SystemChassis instance. The value 17 (Main System Chassis) of the ChassisPackageType property indicates that the represented chassis is a system chassis . As a system chassis , it represents its relationship with the managed system it hosts through a ComputerSystemPackage instance.

The Tag property of the systemchassis1 instance represents the asset tag of the chassis.

The FRUInfoSupported property of the PhysicalAssetCapabilities instance fancapabilities1 indicates by its value of TRUE that the systemchassis1 instance contains FRU information (see FRUInformation).

752



753 **Figure 2 – System chassis object diagram**

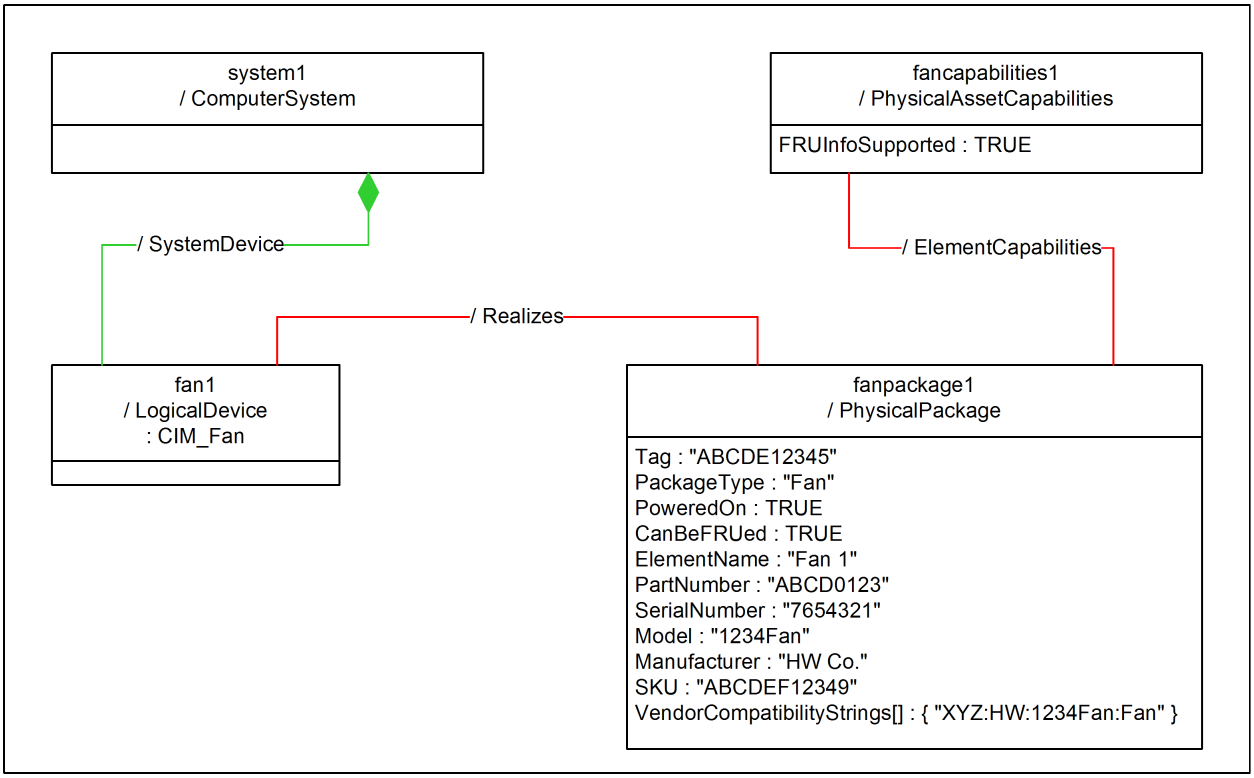
754

8.2 State description: FanPackage

755 The following figure shows a simple object diagram that represents a scenario that conforms to the requirements of this profile. In that scenario, there is a physical package that is a fan and that is represented by the PhysicalPackage instance fanpackage1. The logical aspect of the fan is represented by the LogicalDevice instance fan1 that is associated to the PhysicalPackage instance through Realizes .

756 The FRUInfoSupported property of the PhysicalAssetCapabilities instance fancapabilities1 indicates by its value of TRUE that the fanpackage1 instance contains FRU information (see FRUInformation).

757



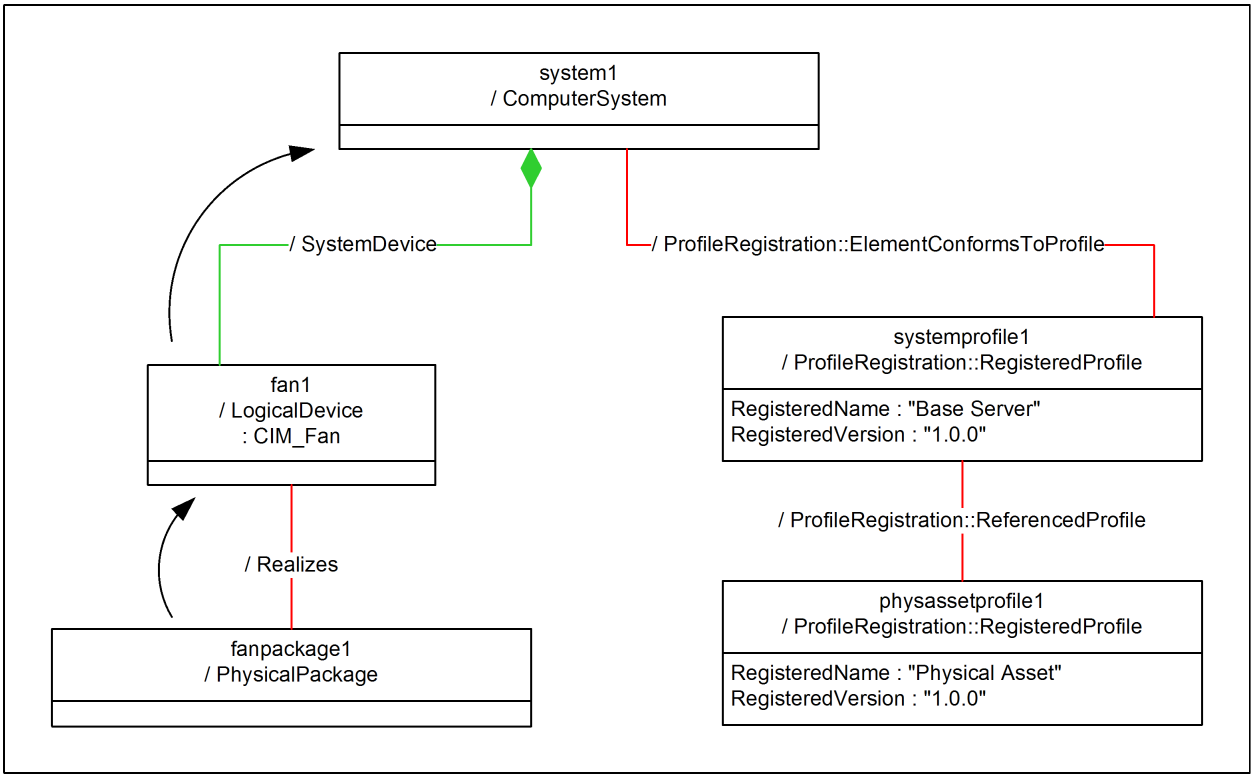
758 **Figure 3 – Fan package object diagram**

759

8.3 Use case: FindScopingInstance1

760 This use case describes the flow to find the scoping instance of a PhysicalPackage instance.

761



762 **Figure 4 – Find scoping instance use case 1**

763

This use case has the following preconditions:

764 • The LogicalElementRelation feature has been implemented for the PhysicalPackage instance.

765 • A PhysicalPackage instance is known.

766 The main flow for this use case consists of the following steps:

767 1. Invoke the Associators( ) for Realizes operation on that PhysicalPackage instance. This will retrieve all LogicalDevice instances representing logical elements of the physical package represented by that PhysicalPackage instance.

768 2. For any one of the returned LogicalDevice instances, determine its scoping instance as described in the Fan Profile. This will result in a ComputerSystem instance.

769 3. Optional: The CIM_RegisteredProfile instance physassetprofile1 representing the implementation of this profile can be found from the scoping instance as described in the Profile Registration Profile.

770

8.4 Use case: FindScopingInstance2

771 This use case describes the flow to find the scoping instance of a PhysicalPackage instance in a physical containment hierarchy.

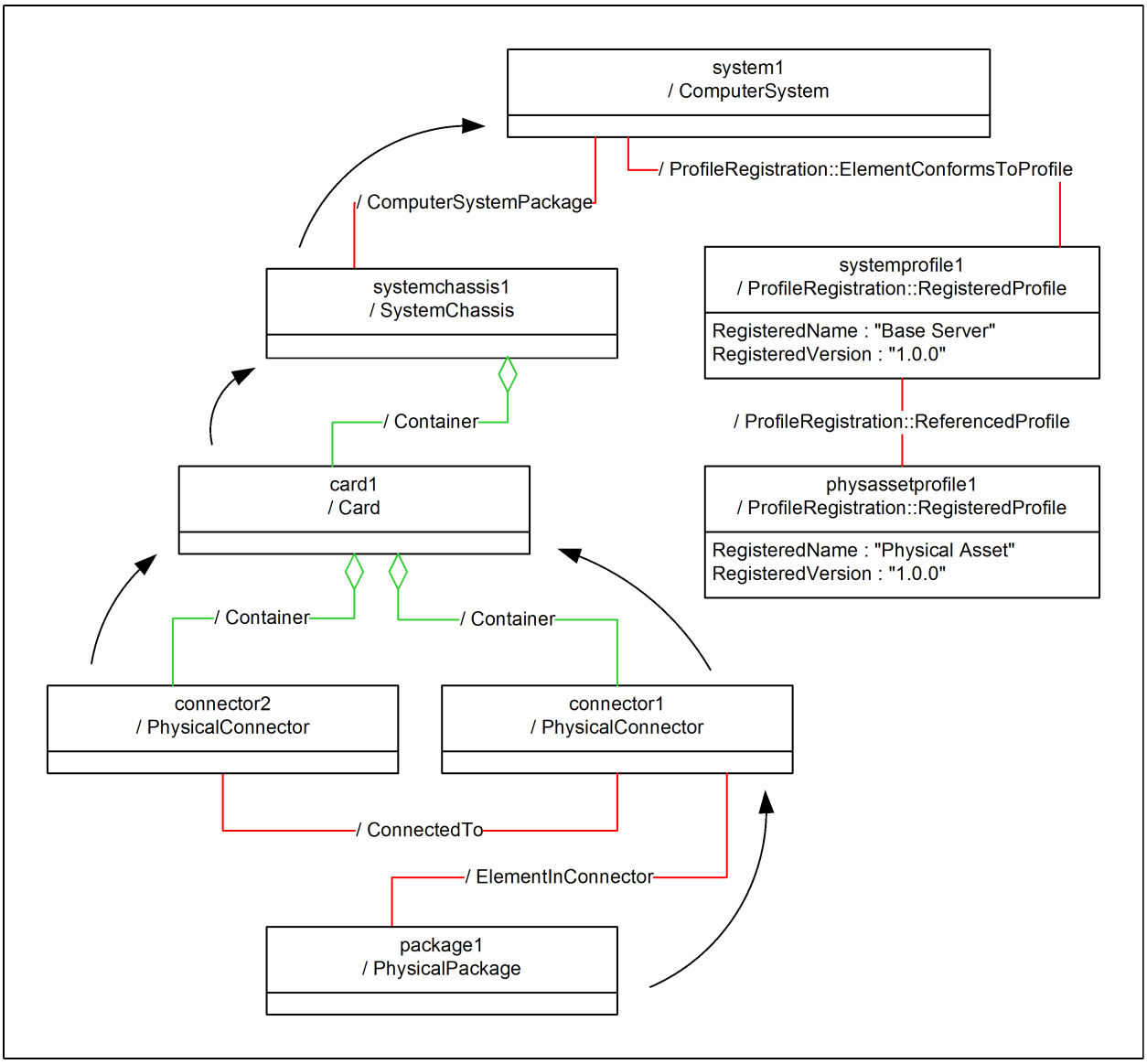


Figure 5 – Find scoping instance use case 2

This use case has the following preconditions:

- The SystemChassisForSystem feature has been implemented for the scoping instance System or the SystemChassisForComputerSystem feature has been implemented for the scoping instance ComputerSystem .
- A PhysicalPackage instance is known.

The main flow for this use case consists of the following steps:

1. Because the PhysicalPackage instance package1 is referenced by the ElementInConnector . Dependent property, the client needs to traverse that association, resulting in the PhysicalConnector instance connector1.

781 2. Because the PhysicalConnector instance connector1 is referenced by the Container . PartComponent property, the client needs to traverse that association, resulting in the Card instance card1.

782 NOTE: To enable finding the scoping instance of connector2, the implementation has instantiated an instance of Container that references card1 and connector2. Merely instantiating the instance of ConnectedTo referencing connector2 will not conform to the scoping class method.

784 3. Because the Card instance card1 is referenced by the Container . PartComponent property, the client needs to traverse that association, resulting in the Chassis instance systemchassis1.

786 4. Because the Chassis instance systemchassis1 is referenced by ComputerSystemPackage , the client needs to traverse that association, resulting in the ComputerSystem instance system1, which is the scoping instance of the PhysicalPackage instance package1.

787 5. Optional: The CIM_RegisteredProfile instance physassetprofile1 representing the implementation of this profile can be found from the scoping instance as described in the Profile Registration Profile.

788

8.5 State description: PhysicalTopology

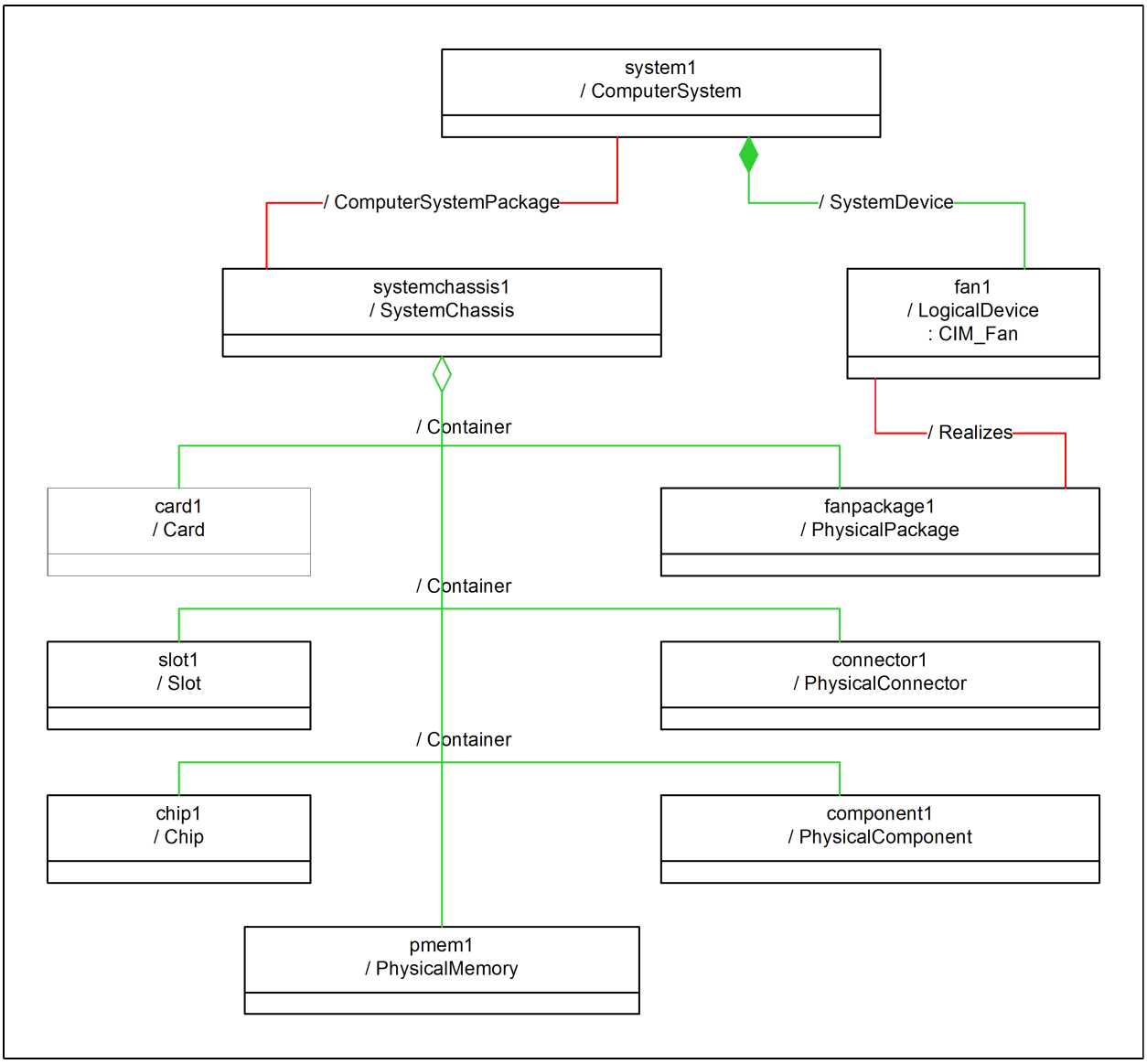
789 The following figure shows an object diagram that represents another possible instantiation of this profile that exposes information about the physical containment hierarchy.

790 The systemchassis1 instance represents the system chassis of the managed system represented by the system1 instance.

791 The fanpackage1 instance represents the physical package of the fan represented by the fan1 instance.

792 The physical topology of the system chassis represented by systemchassis1 contains a single level of hierarchy because the instances card1, slot1, chip1, pmem1, component1, connector1, and fanpackage1 are all directly associated to the systemchassis1 instance through instances of Container .

793



794 **Figure 6 – Physical topology object diagram**

795

8.6 State description: PhysicalMemory

796 The following figure shows an object diagram that represents another possible instantiation of this profile that exposes information about physical memory packaging.

797 The memory of the system represented by the system1 instance is represented by the memory1 instance.

798 The physical aspects of that memory is represented by the pmem1 instance.

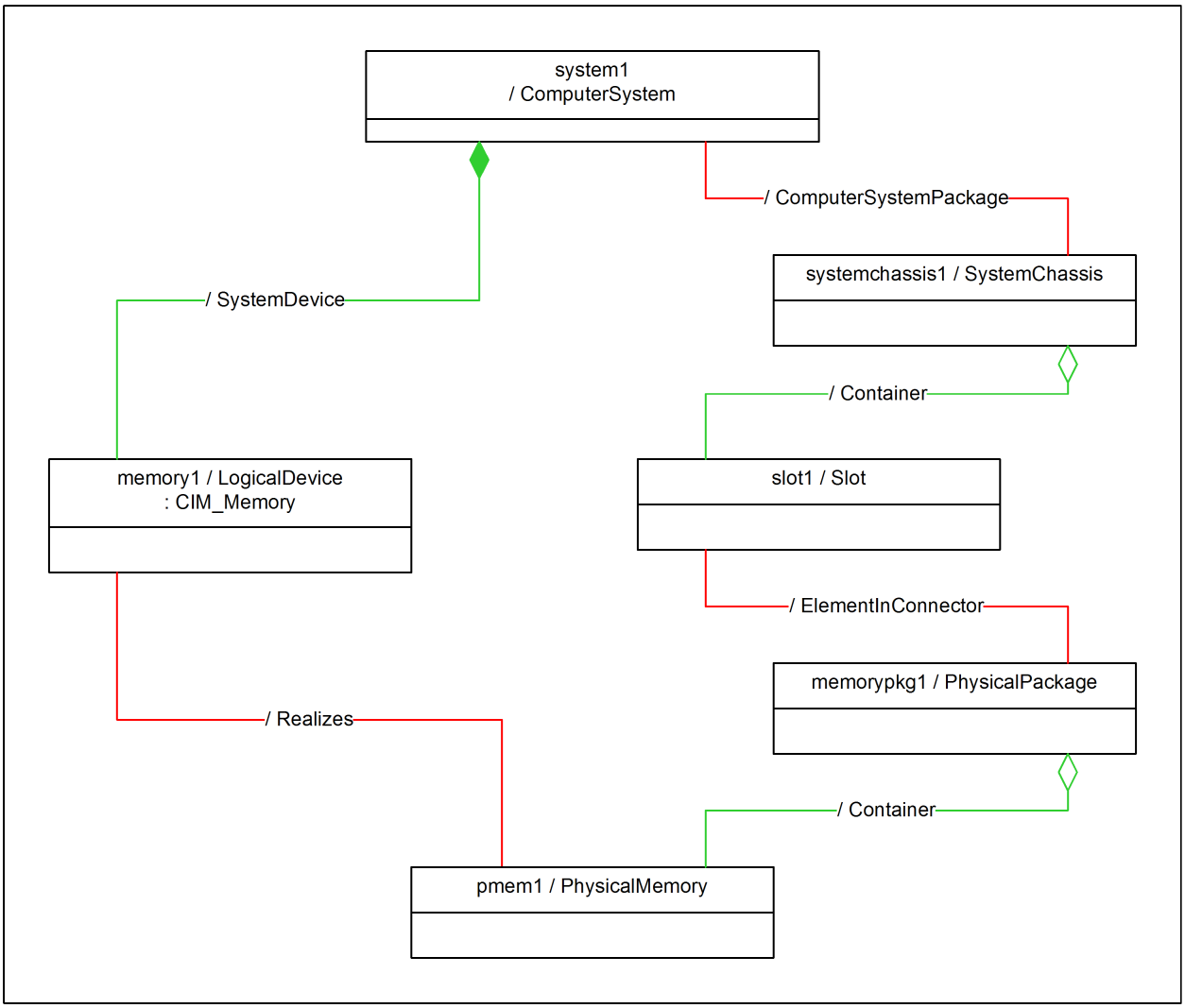
799 The system chassis of the managed system represented by the system1 instance is represented by the systemchassis1 instance.

800 The system chassis represented by the systemchassis1 instance contains a slot represented by the slot1 instance, into which the memory package represented by the memorypkg1 instance is plugged.

801

The memory package represented by the memorypkg1 instance contains the physical component that is the memory chip, represented by the pmem1 instance.

802



803 **Figure 7 – Physical memory object diagram**

804

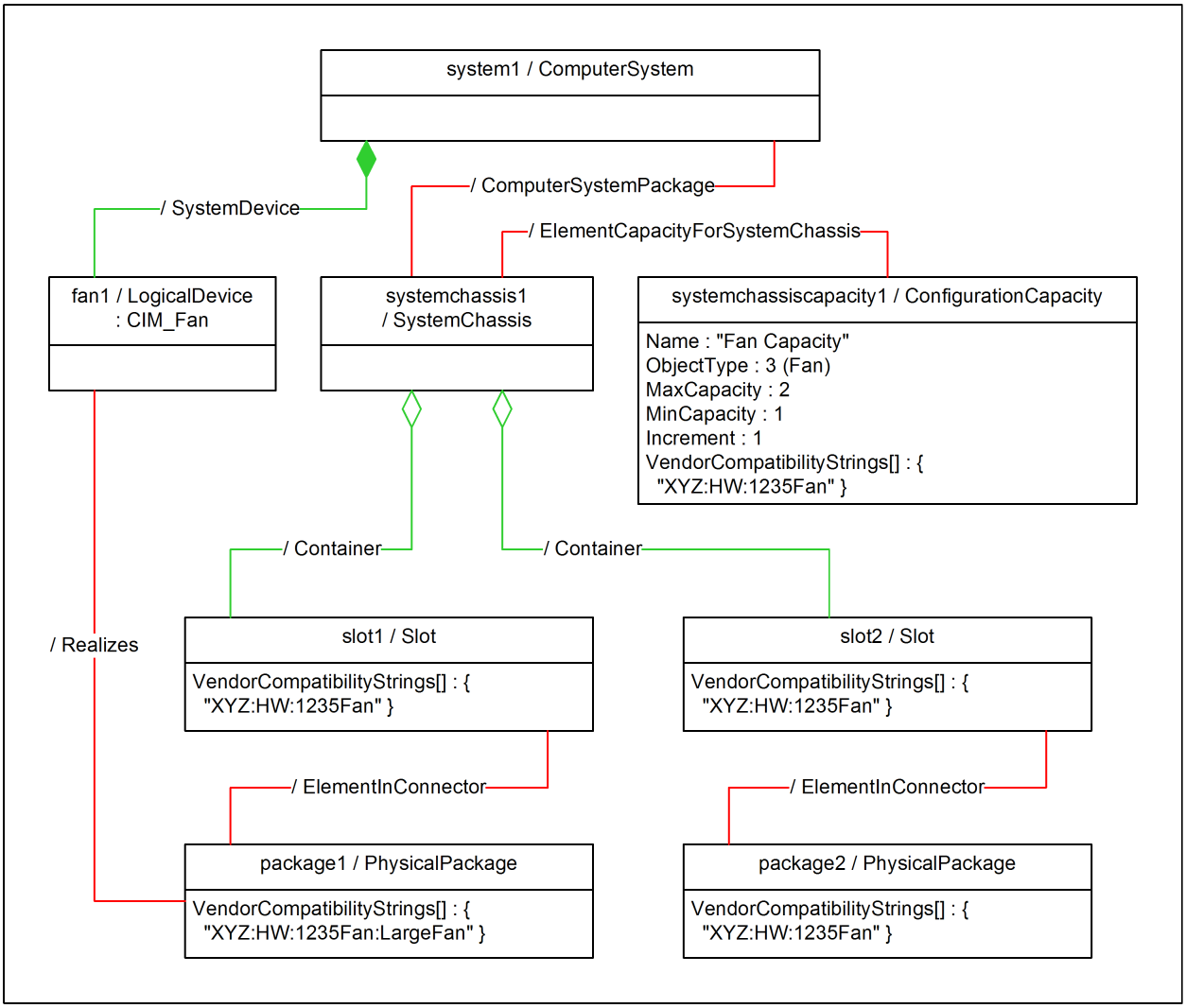
8.7 State description: ConfigurationCapacity1

805 The following figure shows an object diagram that represents another possible instantiation of this profile that exposes information about the configuration capacity of the managed system.

806 Todo: This description so far is a plain copy of the profile text and needs to be updated.

807 In this instantiation, the chassis1 has two slots: slot1 and slot2. The slots are compatible with any type of XYZ:HW:1235Fan packages, as advertised through the CIM_Slot.VendorCompatibilityStrings property. slot1 and package1, which is plugged into it, are compatible because the Delimited Substring matches for the VendorCompatibilityStrings property. slot2 and package2, which is plugged into it, are compatible because an element in the VendorCompatibilityStrings property of the CIM_Slot instance is a Delimited Substring of the element in the VendorCompatibilityStrings property of the CIM_PhysicalPackage instance. chassis1 also has a representation of its fan configuration capacity through capacity1. capacity1 indicates that chassis1 can have a maximum of two fans and should have at least one fan.

808



809 **Figure 8 – Configuration capacity 1 object diagram**

810

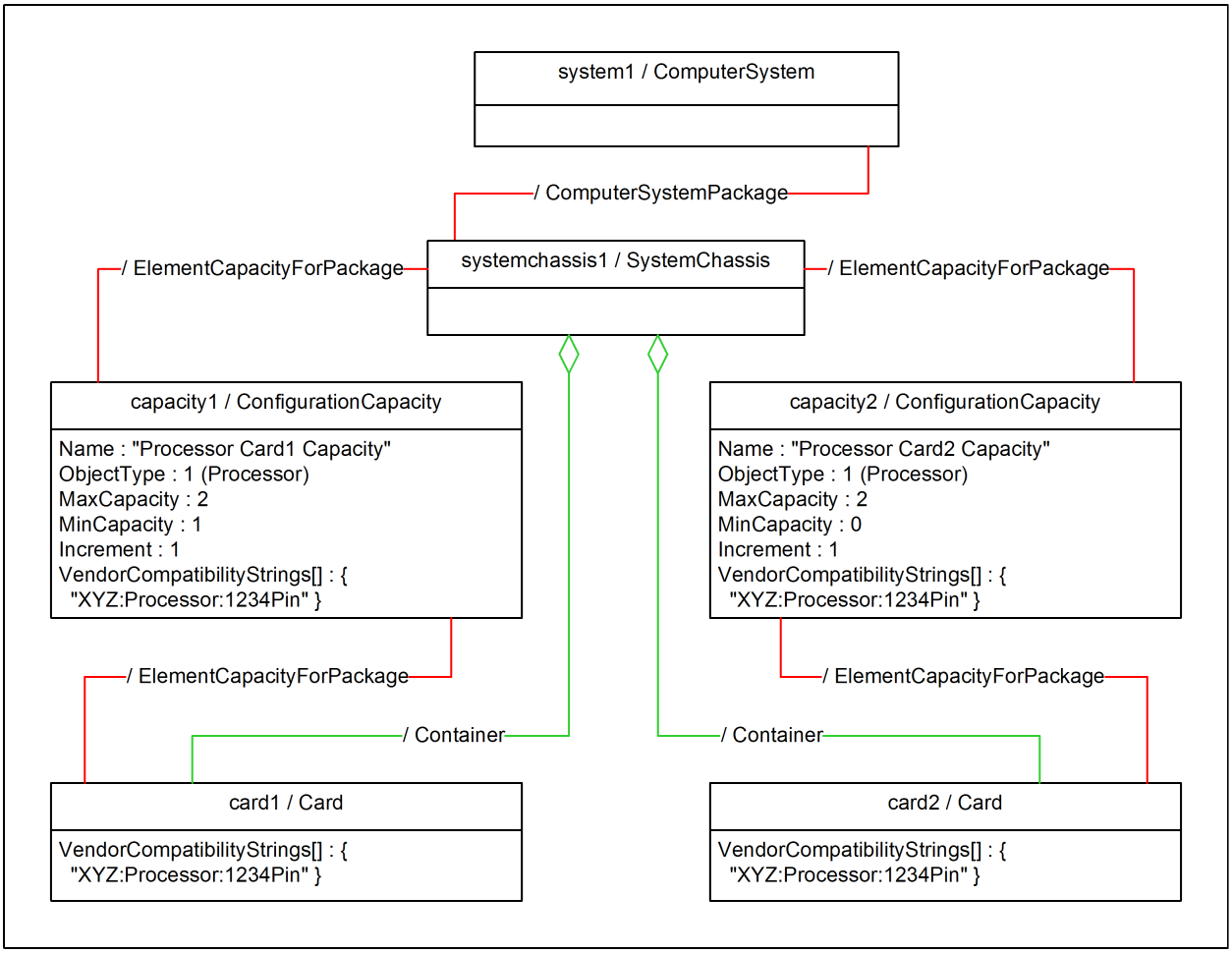
8.8 State description: ConfigurationCapacity2

811 The following figure shows an object diagram that represents another possible instantiation of this profile that exposes information about the configuration capacity of two cards.

812 Todo: This description so far is a plain copy of the profile text and needs to be updated.

813 In this instantiation, the chassis1 has two cards (card1 and card2) that hold processors. The configuration capacity for card1 is represented by capacity1 because they are associated through the instance of CIM_ElementCapacity. In the same way, card2's configuration capacity is represented by capacity2. Because the VendorCompatibilityStrings property value for capacity1 is equal to the VendorCompatibilityStrings property value for capacity2, the maximum number of compatible processors could be determined by adding the MaxCapacity property value of capacity1 to the MaxCapacity property value of capacity2. In this case, the chassis1 could contain a maximum of four processors.

814



815 **Figure 9 – Configuration capacity 2 object diagram**

816

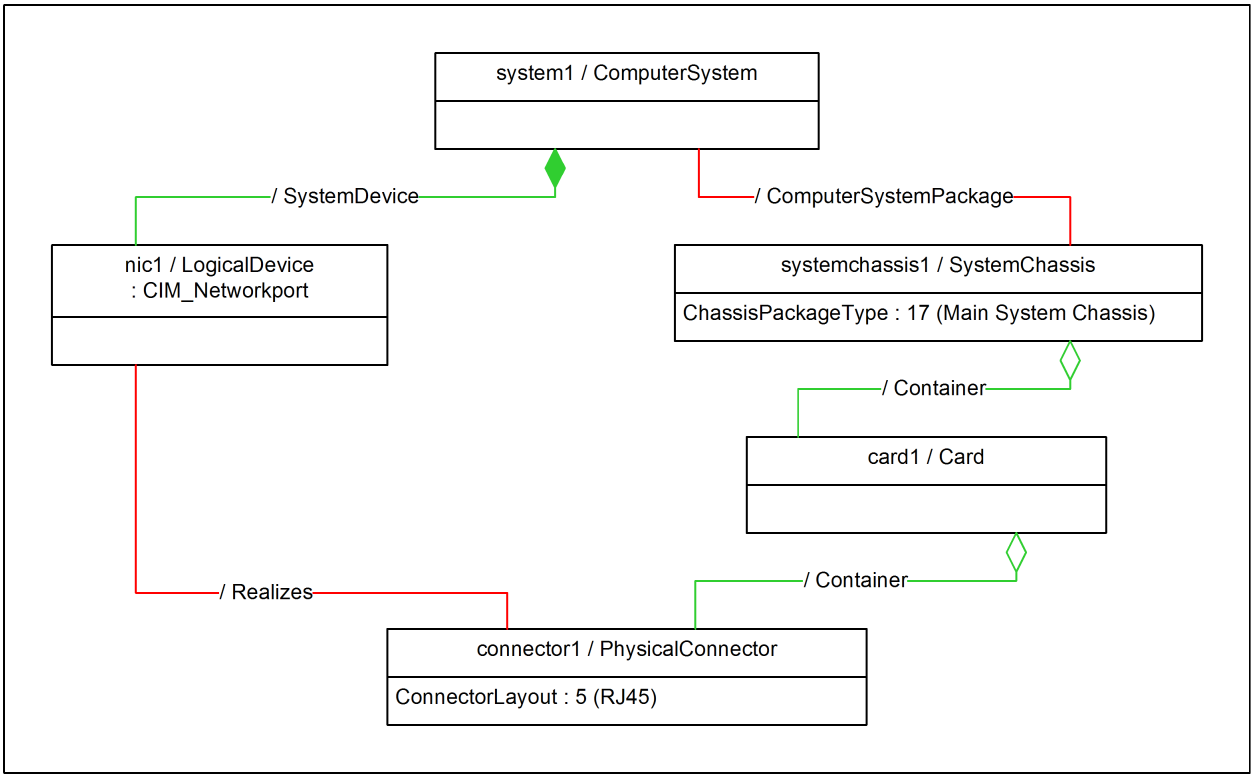
8.9 State description: NetworkPortConnector

817 The following figure shows an object diagram that represents another possible instantiation of this profile that exposes information about the connector of a network card.

818 Todo: This description so far is a plain copy of the profile text and needs to be updated.

819 In this instance, chassis1 contains a network card, card1. card1 has an RJ45 connector, connector1. connector1 is the physical representation of nic1 network port within system1.

820



821 **Figure 10 – Network port connector object diagram**

822

8.10 Use case: DeterminePartNumberOfComponent

This use case describes the flow to determine the part number of a failing physical element.

823 The main flow for this use case consists of the following step:

825 1. Todo: This description so far is a plain copy of the profile text and needs to be updated.

826 Select the CIM_PhysicalElement subclass instance that is associated through the CIM_Realizes association to the CIM_LogicalDevice component that has a HealthState or OperationalStatus property value indicating that the component is in a failure mode. Get the PartNumber property value for the selected CIM_PhysicalElement subclass instance.

827

8.11 Use case: GetPhysicalInventoryOfSystem

This use case describes the flow to obtain the physical inventory for all devices within a system.

828 The main flow for this use case consists of the following step:

830 1. Todo: This description so far is a plain copy of the profile text and needs to be updated.

831 Select the CIM_System instance representing the given system. Select all the CIM_LogicalDevice subclass instances that are associated with the CIM_System instance through the CIM_SystemDevice association, and select all the CIM_System instances associated through CIM_SystemComponent associations, and then follow the CIM_SystemDevice association to select all the CIM_LogicalDevice subclass instances. Get all the property values of the CIM_PhysicalElement subclass instances that are associated to the selected CIM_LogicalDevice subclass instances through the CIM_Realizes association and to the selected CIM_System instances through the CIM_SystemPackage association.

832

8.12 Use case: GetPhysicalInventoryOfSystemChassis

This use case describes the flow to obtain the physical inventory for a system chassis.

The main flow for this use case consists of the following step:

1. Todo: This description so far is a plain copy of the profile text and needs to be updated.

   Get all the property values of the Physical Package instances that are associated through the CIM_SystemPackaging association with the CIM_System instance representing the given system.

8.13 Use case: DetermineSlotIsEmpty

This use case describes the flow to determine whether a slot is empty.

The main flow for this use case consists of the following step:

1. Todo: This description so far is a plain copy of the profile text and needs to be updated.

   Select all the CIM_ElementInConnector instances that reference the CIM_Slot instance that represents the given slot. If no instances of CIM_ElementInConnector that reference the CIM_Slot instance exist, then the slot is empty; otherwise the slot is occupied by the physical package represented by the instance of CIM_PhysicalPackage referenced by the CIM_ElementInConnector association instance.

8.14 Use case: GetFanCapacityOfChassis

This use case describes the flow to retrieve the fan capacity for a chassis.

The main flow for this use case consists of the following step:

1. Todo: This description so far is a plain copy of the profile text and needs to be updated.

   For the CIM_Chassis instance that represents the given chassis, select the associated instances of CIM_ConfigurationCapacity through the CIM_ElementCapacity associations. Select CIM_ConfigurationCapacity instances that have the CIM_ConfigurationCapacity.ObjectType property of 3 (Fan).

8.15 Use case: GetMaxFanCapacityOfChassis

This use case describes the flow to retrieve the maximum capacity of the type of fan package within a chassis.

The main flow for this use case consists of the following step:

1. Todo: This description so far is a plain copy of the profile text and needs to be updated.

   The particular type of fan package is identified through the given string, which is an element of the VendorCompatibilityStrings array property of the Physical Package representing the fan package. Select all the instances of CIM_ConfigurationCapacity associated with the CIM_Chassis instance through instances of CIM_ElementCapacity where the VendorCompatibilityStrings array property of the instance of CIM_ConfigurationCapacity contains elements equal to the given string. Add all the values for the MaxCapacity property of the selected CIM_ConfigurationCapacity instances.

ANNEX A
(informative)

Change log

853

Version	Date	Description
1.0.0b	2006-06-28	DSP1011: Preliminary Standard
1.0.0	2007-11-12	DSP1011: Final Standard
1.0.1	2008-06-09	DSP1011: Incorporated errata submitted for the Final Standard.
1.0.2	2009-06-04	DSP1011: DMTF Standard Release. Incorporated errata on CIM_PhysicalMemory.Speed property values for unknown or variable speeds.
1.0.3m	2011-08-31	XMP1011: Included as a sample profile into DSP2023

854

Bibliography

This clause lists references that are helpful for the application of this document.

DMTF DSP1000, *Management Profile Specification Template 1.1*, http://www.dmtf.org/standards/published_documents/DSP1000_1.1.pdf