

ソフトウェア向けハードウェアモデルの 重要性和SHIM2.0

2017-11-16

Masaki Gondo, eSOL

SHIM WG Chair

Agenda

- 何故ソフトウェア向けのハードウェアモデル記述なのか
- SHIM
- まとめ

何故ソフトウェア向けのハードウェアモデル記述なのか

ソフトウェアの高度化

海外の動向

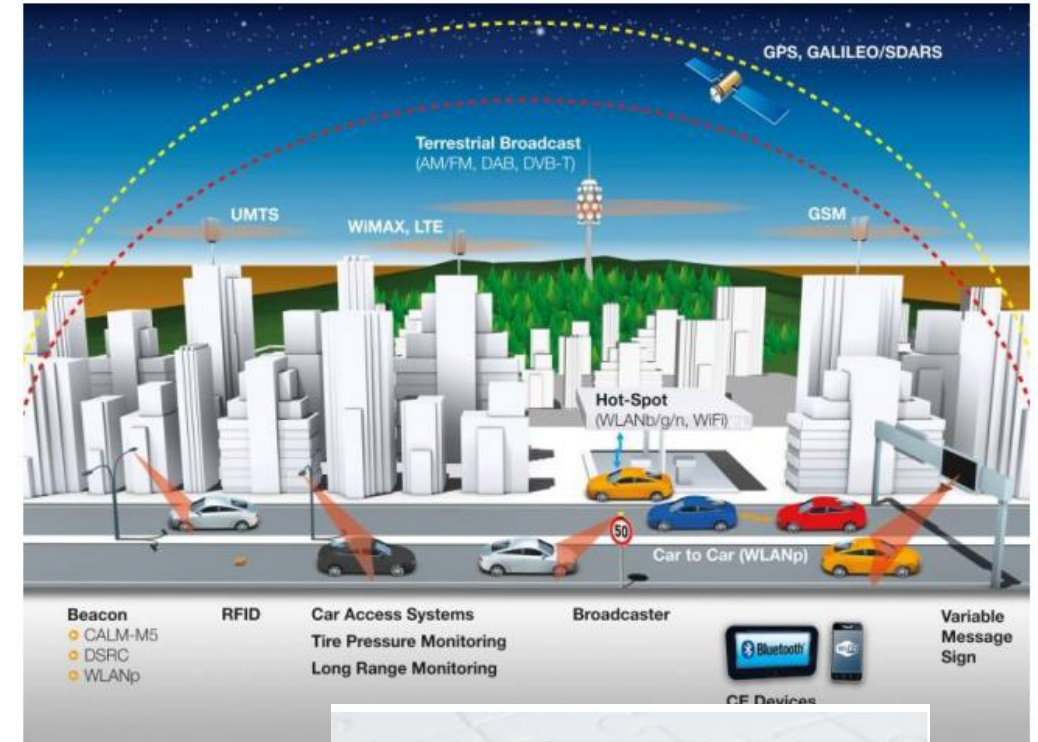
ソフトウェアの高度化と高性能なハードウェア

- システム要求からの流れ
 - システムのソフトウェア化→ソフトウェア高度化→高い性能要求
 - 制約：ムーアの法則の終焉
- 高度なソフトウェアでコンピュータとして高い性能を実践的な開発コストで実現するためには、**高性能なHWに最適化するためのメソドロジが不可欠**
- **メソドロジ（ツール）はハードウェアを知らなければならない**
→ハードウェアモデルの必要性

landscape of intelligent ECUs

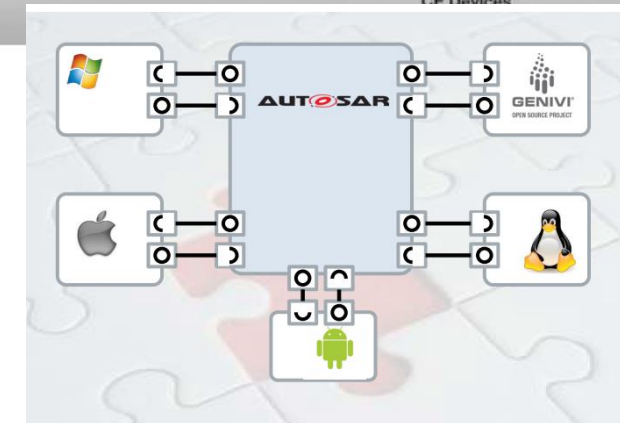
Highly automated driving will be on the road.

HAD



Use cases

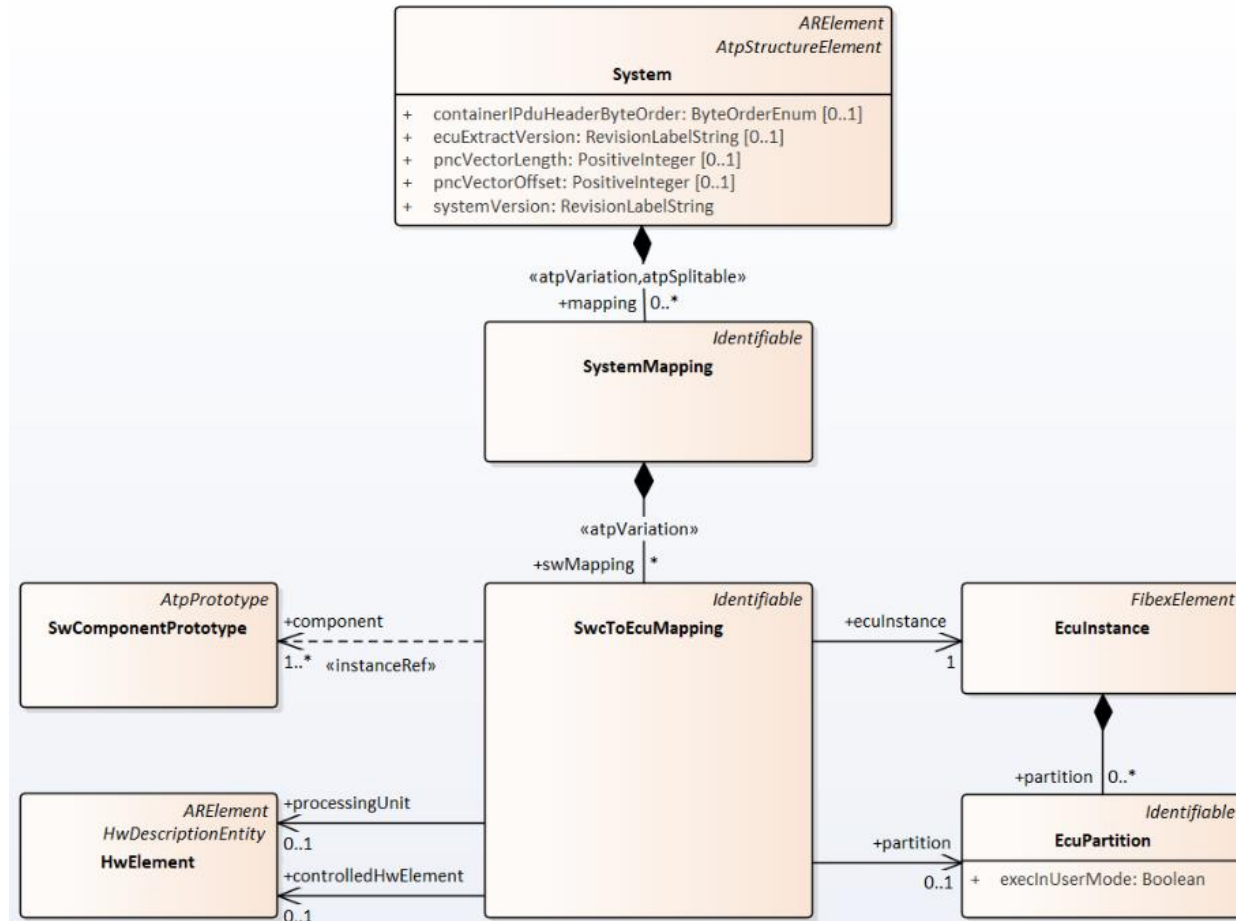
- Support dependable systems including fail-operational systems
- Support of cross domain computing platforms
- Support of high-performance micro-controllers and computing
- Distributed and remote diagnostics
- ...



Technology Drivers for AUTOSAR Adaptive Platform

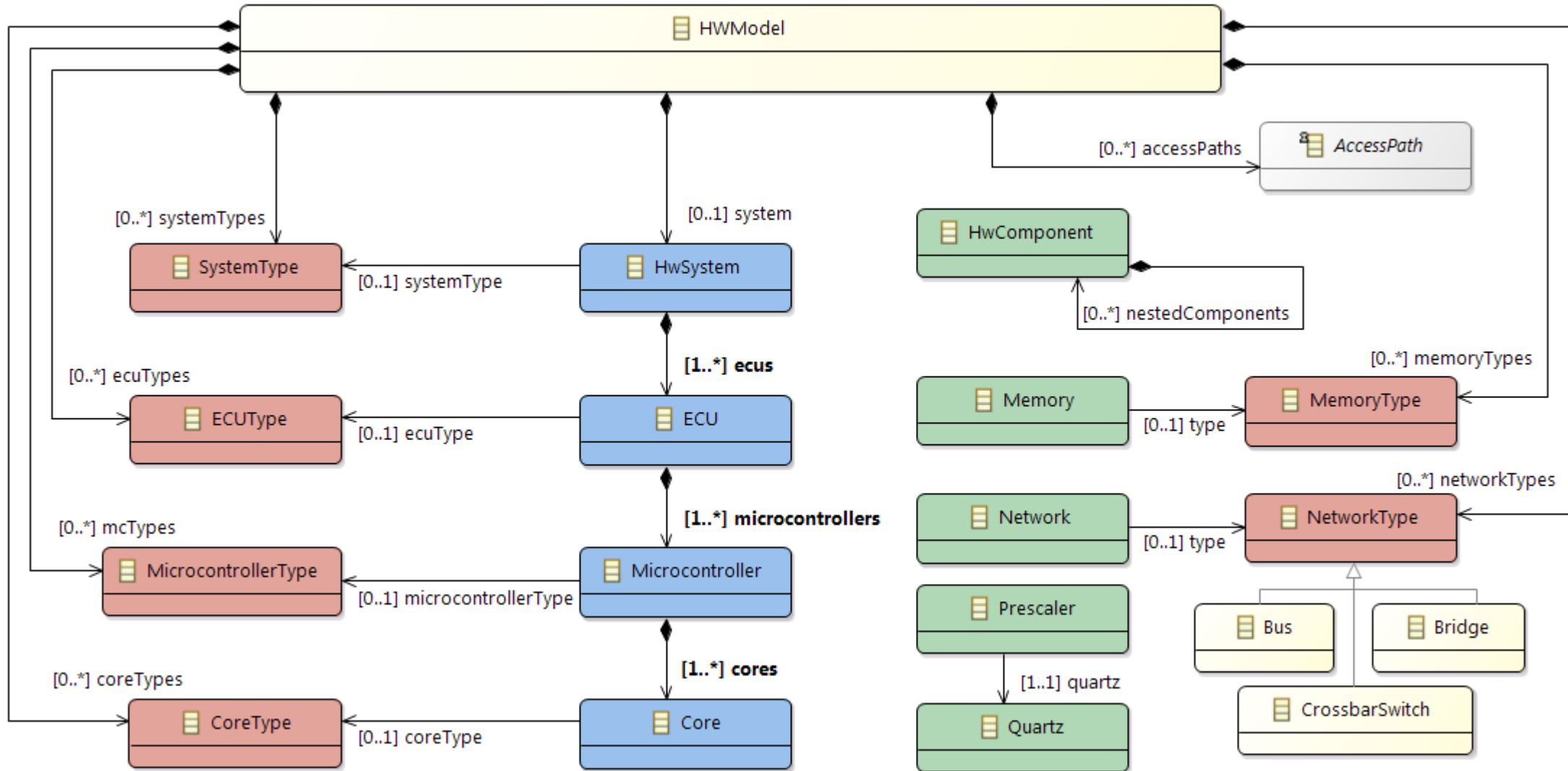
- HADを始めとするシステムの実現性に不可欠なIntelligent ECUの実現には**新たな技術**が求められる
- AIを始めとする様々な新しい処理には、**多くのデータと多くの演算**を高速に処理できる技術が必要となる
- 具体的には車載ネットワークはCANから**Ethernet**へ、計算はマイコンからプロセッサによる**コンピューティング**へ
- これは**組込みにおけるHPC** (High Performance Computing) と言える
- マルチコアは最低限、**8コアを優に超えるメニーコア、GPGPU、FPGA、DFP、専用アクセラレータ等を組合わせたヘテロジニアスコンピューティング**の導入がすでに始まっている

AUTOSAR Classic Platformのハードウェアモデル



- SW ComponentのECUへのマッピングとパーティションへのマッピング
- マッピングのための簡易的なもので性能情報などは存在しない
- ただし、HWとSWはセットであるということを示している一例

AMALTHEAのハードウェアモデル



<https://www.eclipse.org/app4mc/help/app4mc-0.7.2/index.html#section3.7>

AMALTHEAのハードウェアモデル詳細



CoreType

Microcontroller

The *Microcontroller* is a specialization of the *ComplexNode* and represents processors with one or more special or generic purpose cores.

Attribute	Type	Mul.	Valid Values	
microcontrollerType	Reference	1	<i>MicrocontrollerType</i>	Refers to microcont
cores	Containment	+	<i>Core</i>	Containm microcont

The *CoreType* is a specialization of the *AbstractionTypeDescription* and used to define a type of a Core, i.e. its bit width and ticks per cycle.

Attribute	Type	Mul.	Valid Values	Description
bitWidth	Integer	1	0-255	The cores architectures bit width (e.g. 32 or 64 bit)
instructionsPerCycle	Integer	1	0-255	Number of processed (computed) instructions per cycle.

MemoryType

Core

The *Core* is a specialization of the *ComplexNode* and represents the respe processor.

Attribute	Type	Mul.	Valid Values	
coreType	Reference	1	<i>CoreType</i>	Refers to a <i>CoreType</i> conta
lockstepGroup	Integer	?	0 – 255	If set, this value defines the group are operating in locks

The *MemoryType* is a specialization of the *HardwareTypeDescription* and used to define a type of a memory, i.e. its size, access pattern and type.

Attribute	Type	Mul.	Valid Values	Description
xAccessPattern	String	1	String	Timing Architects specific value. Will be used in the future to determine the access pattern of a memory element.
type	<i>MemoryType</i>	1	RAM/CACHE/FLASH_INT/FLASH_EXT	Specifies the type of this Memory.
size	Long	1	>0	Size of this memory.

Memory

The *Memory* is a specialization of the *ComplexNode* and represents memo RAM, etc.

Please note that to specify a memory with the type *CACHE* (e.g. lv1 cache necessary to specify additional networks or ports.

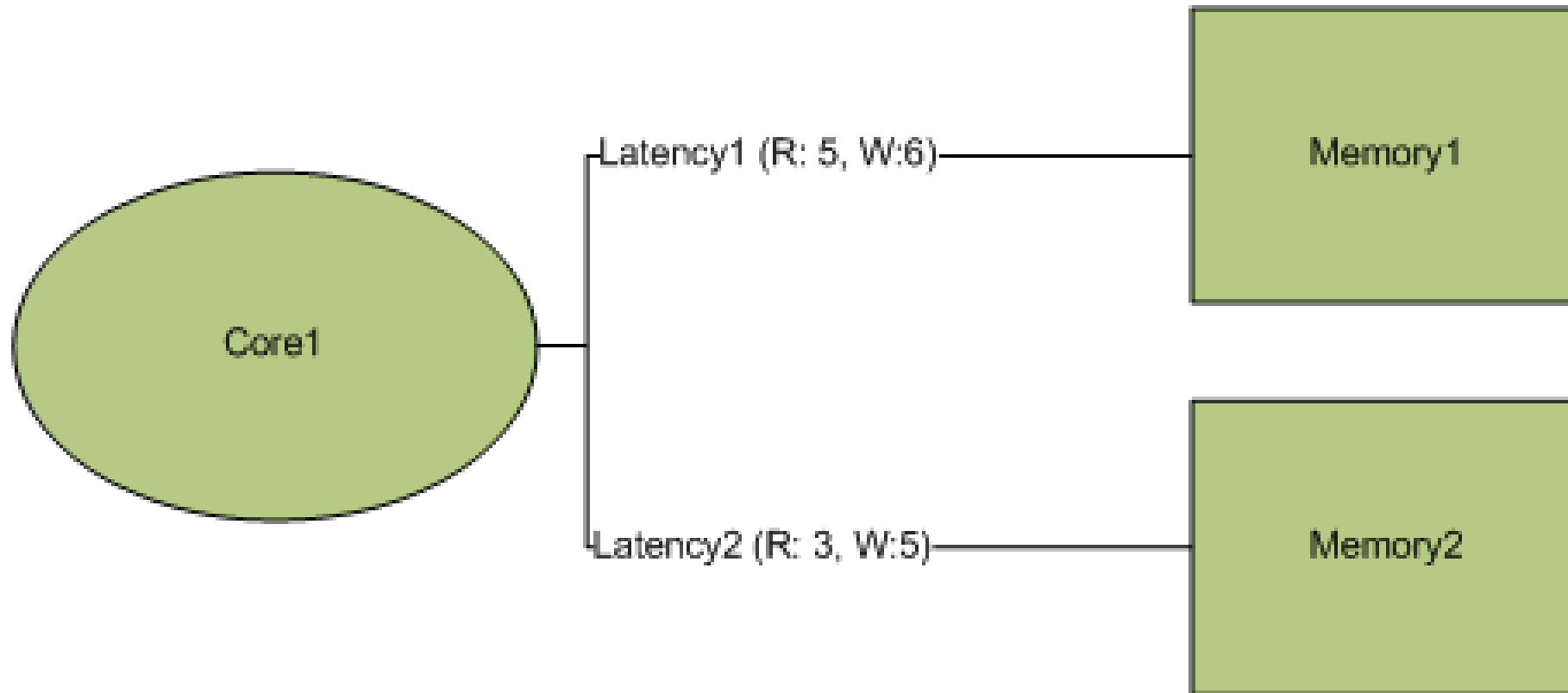
Attribute	Type	Mul.	Valid Values	
type	Reference	1	<i>MemoryType</i>	Refers to a <i>MemoryType</i> con

NetworkType

The *NetworkType* is a specialization of the *HardwareTypeDescription* and used to define a type of a Network, i.e. max supported bit width and scheduling policy.

Attribute	Type	Mul.	Valid Values	Description
schedPolicy	<i>SchedType</i>	1	RROBIN/PRIORITY	Specifies the scheduling policy which is used on this Network.
bitWidth	Integer	1	0 – 255	Specifies the max bit width that may be transferred on this network per cycle.

AMALTHEAのメモリアクセス性能情報



SHIM

The Multicore Association

- 2005設立
- ミッション: 業界標準化によってTime-To-Marketを改善
- 委員会ベースの規格策定



SHIM

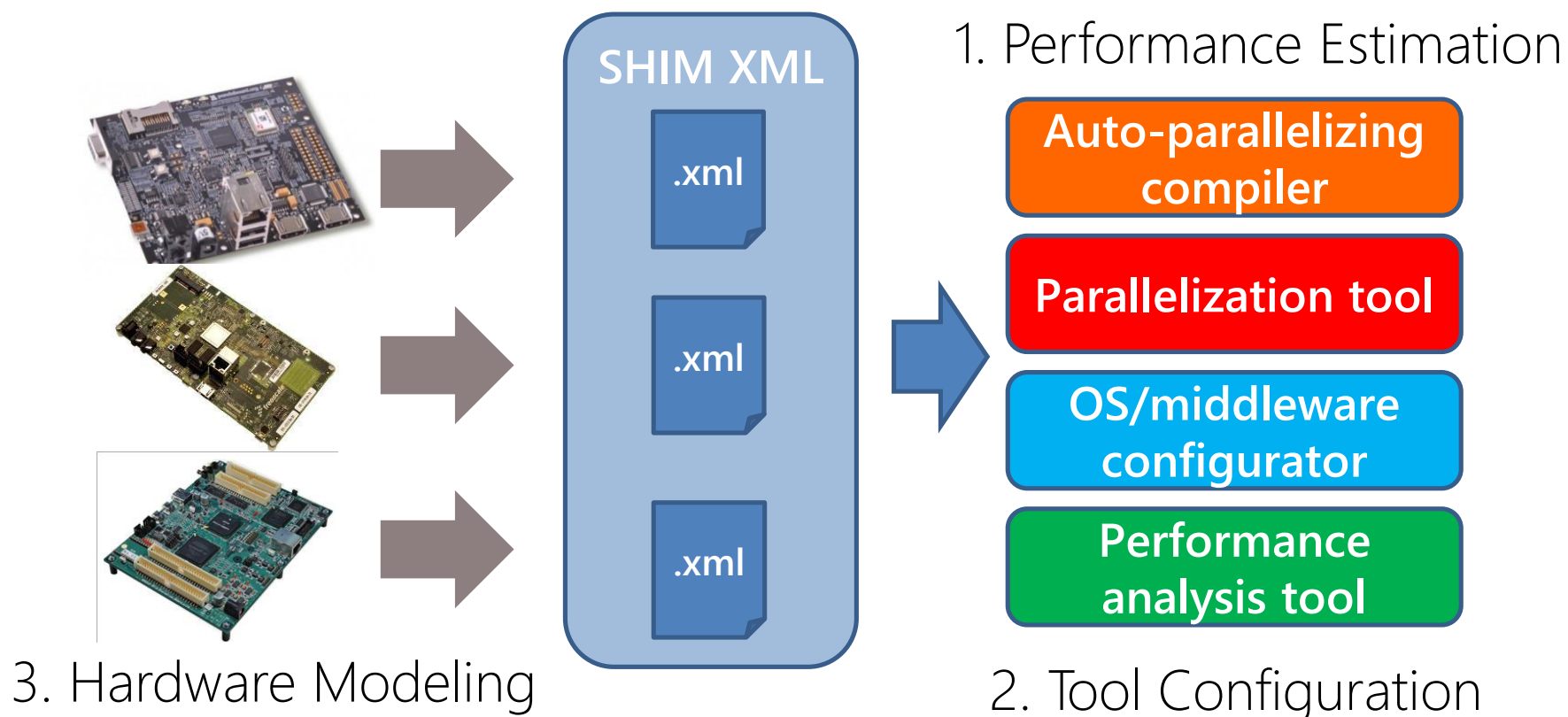
Shim (spacer)

From Wikipedia, the free encyclopedia

A **shim** is a thin and often tapered or **wedged** piece of material, used to fill small gaps or spaces between objects.^[1] Shims are typically used in order to support, adjust for better fit, or provide a level surface.

Shims may also be used as spacers to fill gaps between parts subject to wear.

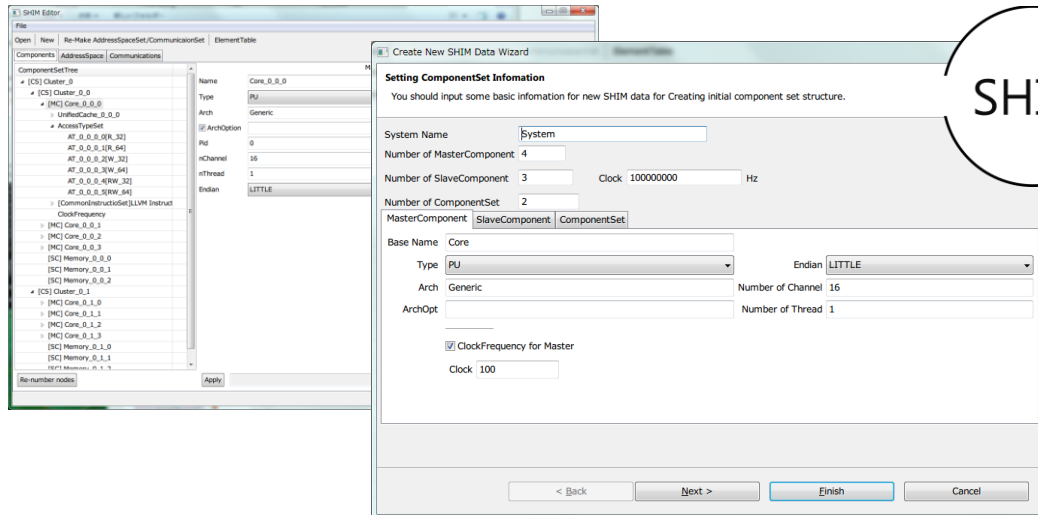
- **S**oftware **H**ardware **I**nterface for **M**ulti-manycore
- ハードウェアを表現するXMLスキーマ標準仕様



SHIM 1.0

- 2015年1月に公開済み
- [MCA Webページ](#)からDownload可能
- github上にSHIM XMLを作成するためのツール群をMITライセンスでOSSとして公開済み

<https://github.com/openshim/shim>

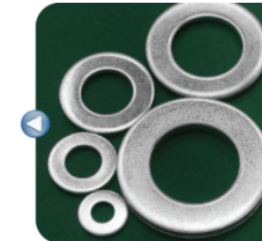


SHIM



Industry Standards to Solve Multicore Challenges
The Multicore Association (MCA) is an industry association that includes leading-edge companies implementing products that embrace multicore technology. Our primary objective is to define and promote open specifications to enable multicore product development.

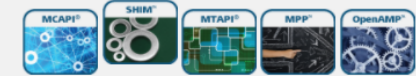
HOME ABOUT FAQ WORKING GROUPS MEMBERSHIP NEWS RESOURCES



SHIM™ Software-Hardware Interface for Multi-many-core

An architecture description standard from the software design perspective

Provides a common interface that abstracts the hardware properties that are critical to enable multicore tools.



NEW FROM THE MULTICORE ASSOCIATION

Video Describing Multicore Task Management (MTAPI)
Multicore Programming Practice Guide in Japanese の日本語版ができました

MULTICORE ASSOCIATION WORKGROUPS

The **Open Asymmetric Multi Processing Framework (OpenAMP)** working group is standardizing the specification and will have an open source release by October/November. The OpenAMP framework manages systems comprised of multiple operating systems and compute elements. The OpenAMP Framework allows application developers to leverage the parallelism offered by both homogeneous and heterogeneous multicore systems. [Learn more about OpenAMP](#)

The **Multicore Communications API (MCAPI®)** working group is finalizing MCAPI Version 3.0, which also includes official subsets. New features include 'zero copy' functionality, bidirectional interaction between 'application and application' using shared memory and bidirectional interaction between 'application and driver'. [Learn more about MCAPI](#)

The **Multicore Programming Practices (MPP™)** is a multicore software programming guide for the industry that aids in improving consistency and understanding of multicore programming issues. Interested in contributing to the next version of this document? [Learn more about MPP](#)

The **SHIM™** Working Group. SHIM is the Software-Hardware Interface for Multi-Many Core. It provides a common interface to abstract the hardware properties that matter to multicore tools. Interested in contributing to the next version of this specification? [Learn more about SHIM](#)

DOWNLOAD AVAILABLE SPECIFICATIONS

The Multicore Communication API (MCAPI®)

- [Download the MCAPI Specification](#)
- [View the MCAPI reference card](#)
- [Learn more about the MCAPI workgroup](#)

The Software-Hardware Interface for Multi-Many Core (SHIM™)

- [Download the SHIM Specification](#)
- [Learn more about the SHIM workgroup](#)

The Multicore Programming Practice Guide (MPP™)

- [Download the MPP Guide](#)

Heterogeneous Technologies Power Next Generation Embedded Systems

Discussion topics:

- Benefits of heterogeneous architectures
- New standards and approaches for heterogeneous systems development
- Security and safety considerations

[Watch Now >](#)

NEWS AND PRESS

NEWS: MPP, OpenAMP, SHIM Updates [read](#)

NEWS: API Paves Road for Multicore SoCs [read](#)

NEWS: Get Happy with MTAPI [read](#)

NEWS: EMB2 1.0 unterstützt parallele Programmierung heterogener Systeme [read](#)

NEWS: Embedded Multicore Building Blocks Annexes MCA [read](#)

NEWS: Open-Source-Toolset unterstützt Entwickler bei der Multicore-Programmierung [read](#)

NEWS: Open source tools set to help parallel programming of multicores [read](#)

PRESS: Enhanced Open Source Framework Available for Parallel Programming on Embedded Multicore Devices [read](#)

WEBINAR: Upcoming June 20th: Leveraging the OpenAMP Framework for Heterogeneous Software Architecture [read](#)

THE MCA FOUNDATION

The Multicore Association's foundations consist of an extensive set of application programming interfaces (APIs) that support inter-process communications, resource management, and task management. These APIs enable services and functions such as load balancing, power management, and support for the Internet of Things (IoT). The MCA also produces a specification for a common interface to between multicore/manycore hardware and software tools. In addition, the consortium provides a collaboratively-developed multicore software programming guide that aids in improving consistency and understanding of multicore programming issues.

IMPLEMENTATIONS



Mentor Embedded MCAPI®
MCA Board Member



Poly-Messenger/MCAPI®
MCA Board Member



Has multiple SubSpaces, containing MasterSlaveBindingSet and PerformanceSet expressed in terms of Latency and Pitch in cycles

3. CommunicationSet

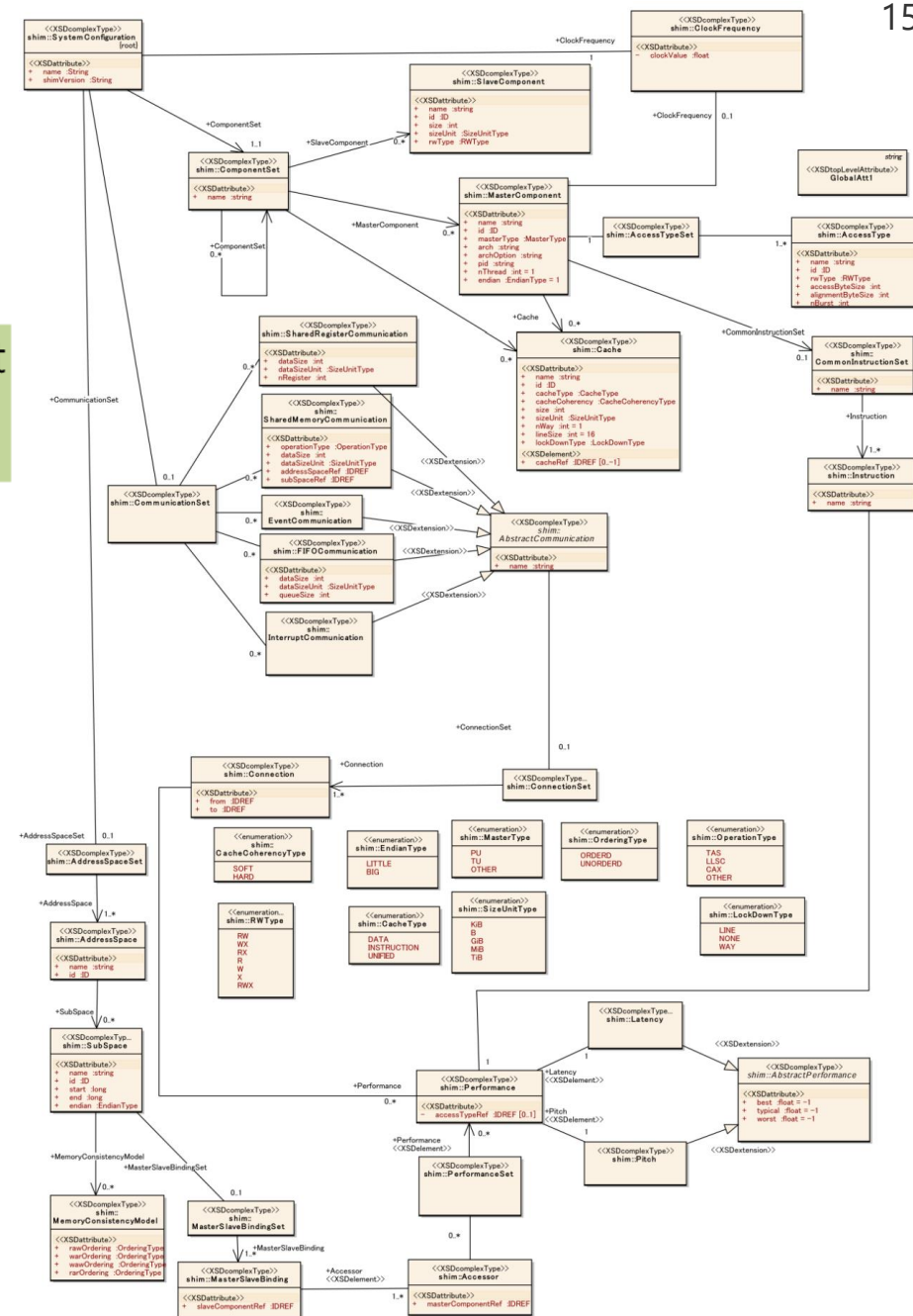
Inter-core comm., has
PerformanceSet containing Latency
and Pitch in cycles

MasterComponent:
Cache, ClockFrequency,
CommonInstructionSet, Access
TypeSet

SlaveComponent: size, RWType

1. ComponentSet

Nest itself, can contain multiple, but single root



SHIM2.0の方向性

1. **Functional units** for improved accuracy
2. **Power** properties with **DFS** support
3. **Complex caches**
4. **Communication contention** description
5. Processor model in a **separate XML for re-use**
6. Vendor **extensions**

SHIMのメリット

- ツール、ランタイム、ハードウェアベンダ
 - **今は**多様なハードウェアとコンフィギュレーションに**各社が個別対応**
 - SHIMにより**多様なハードウェアの対応コスト**が下がる
- システム設計者
 - SHIMによって**早期の性能を含むシステムアーキテクチャ評価**を実ハードウェアの有無によらず行う事が可能になる
- ソフトウェア開発者
 - **SHIM対応ツール**を用いる事で、複雑な**マルチコアハードウェア**を活用しやすくなる

まとめ

- コンピュータの原理
 - **優れたコンピュータはHWとSWが合わせ込まれている**
- ソフトウェアの高度化
 - 高度化したソフトウェアは複雑かつ規模大、開発コスト低減のため**抽象化による再利用→HWへの合せ込みが課題**
- 高性能と高消費電力の二難
 - 高度化したSWを実行する**複雑度の高いヘテロジニアス・マルチ・メニーコア**の必要性
- これらを解決するための**ハードウェアモデルSHIMとSHIM対応設計支援ツール**