ソフトウェア向けハードウェアモデルの重要性と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

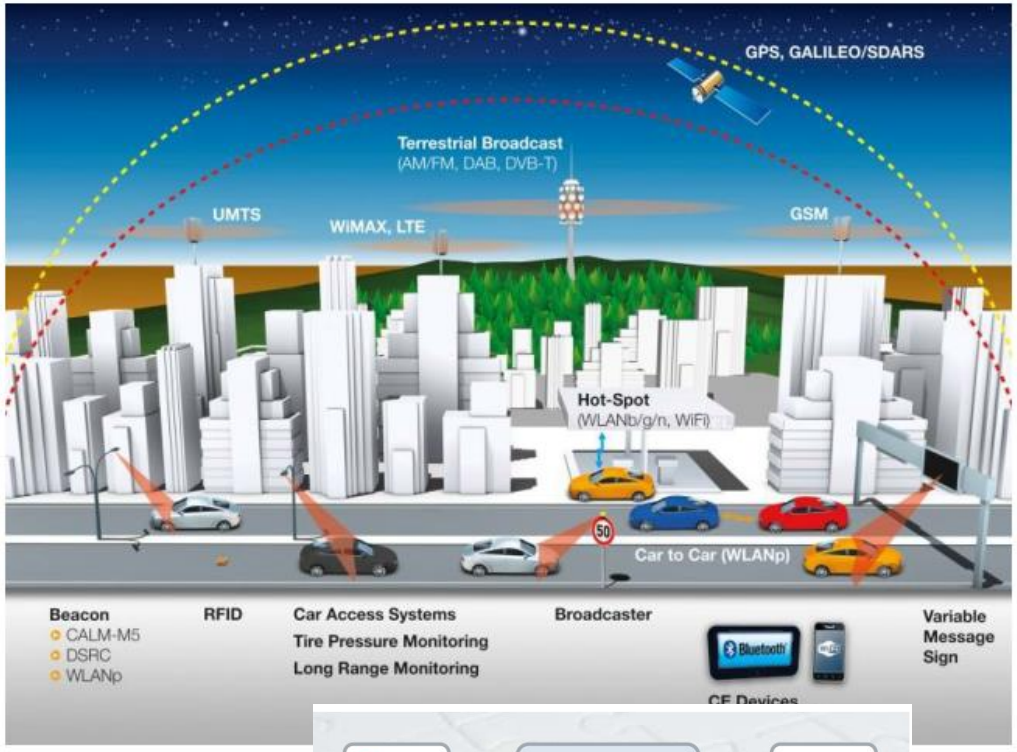
ソフトウェアの高度化と高性能なハードウェア

- 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

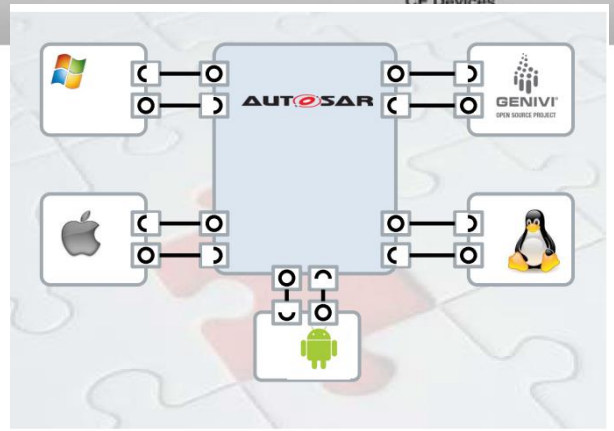
Highly automated driving will be on the road.

HAD



Use cases

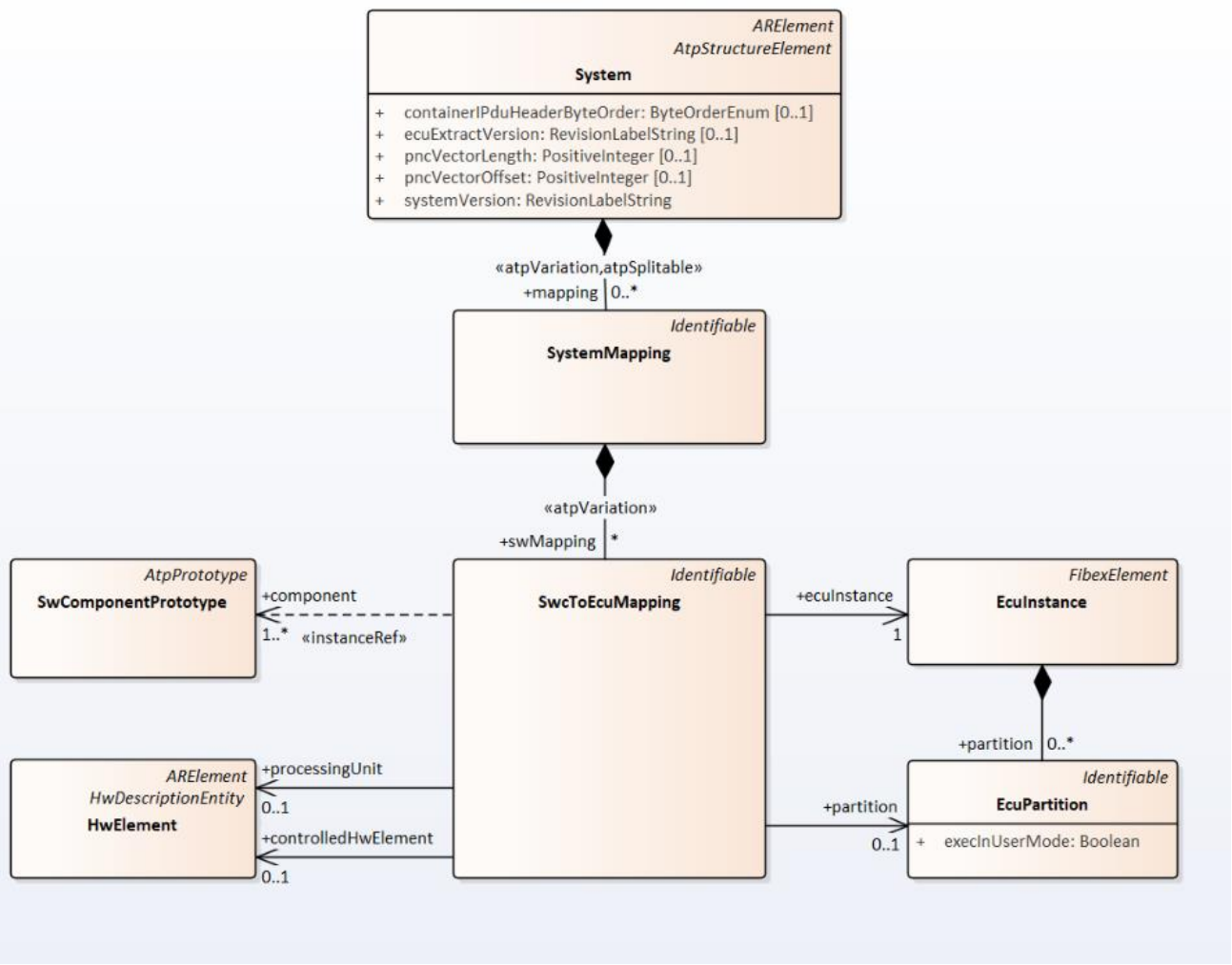
- Support dependable systems including fail-operational systems
- Support of cross domain computing platforms
- Support of high-performance micro-controllers and computing
- Distributed and remote diagnostics
- …



Technology Drivers for AUTOSAR Adaptive Platform

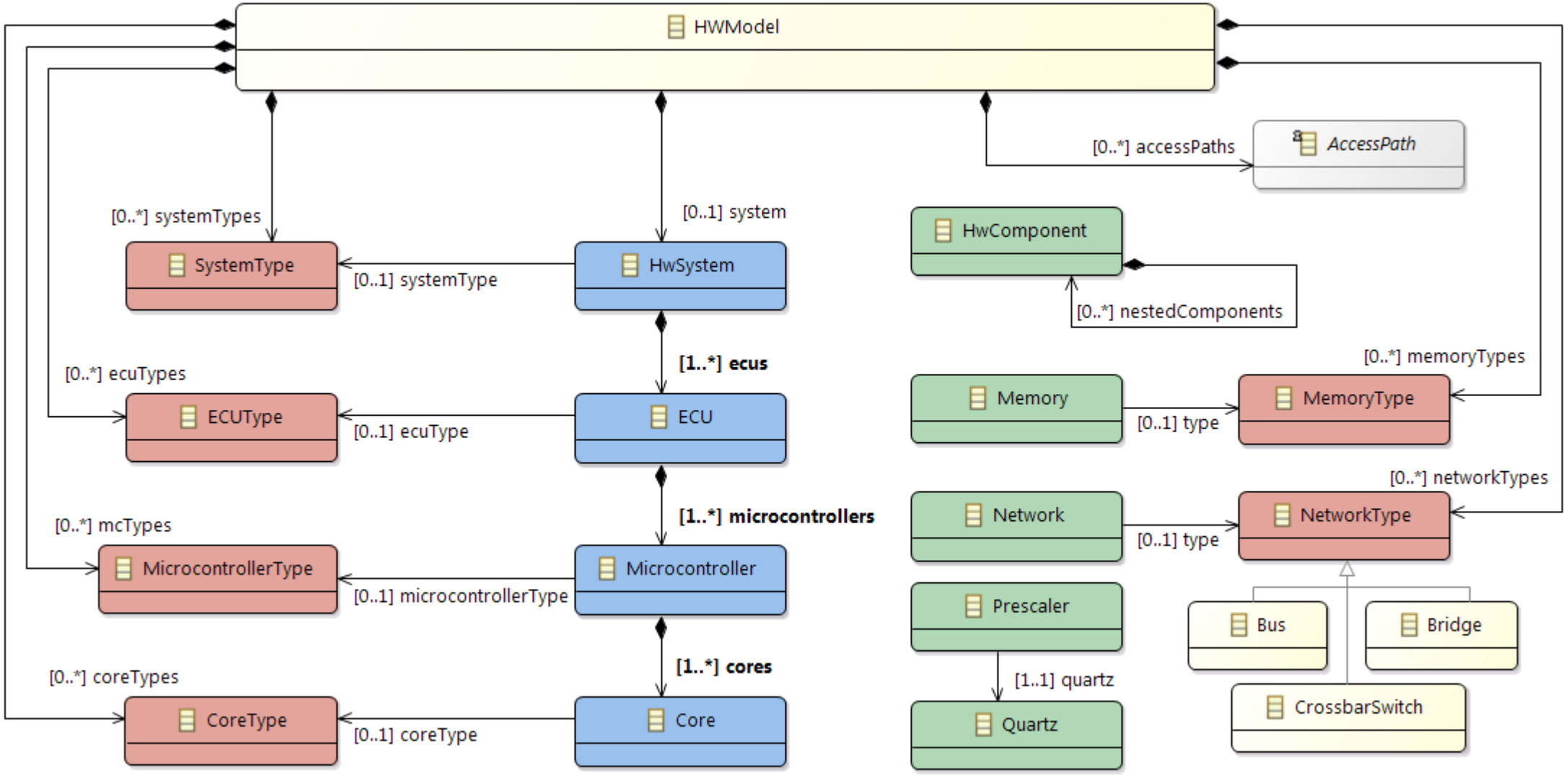
- HADを始めとするシステムの実現性に不可欠なIntelligent ECUの実現には**新たな技術**が求められる
- AIを始めとする様々な新しい処理には、**多くのデータと多くの演算**を高速に処理できる技術が必要となる
- 具体的には車載ネットワークはCANから**Ethernet**へ、計算はマイコンからプロセッサによる**コンピューティング**へ
- これは**組込みにおけるHPC** (High Performance Computing) と言える
- マルチコアは最低限、**8コアを優に超えるメニーコア、GPGPU、FPGA、DFP、専用アクセラレータ等を組合わせたヘテロジ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

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	*MicrocontrollerType*	Refers to microcont
cores	Containment	+	*Core*	Containm microcont

Core

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

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

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	*MemoryType*	Refers to a *MemoryType* con

CoreType

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

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.

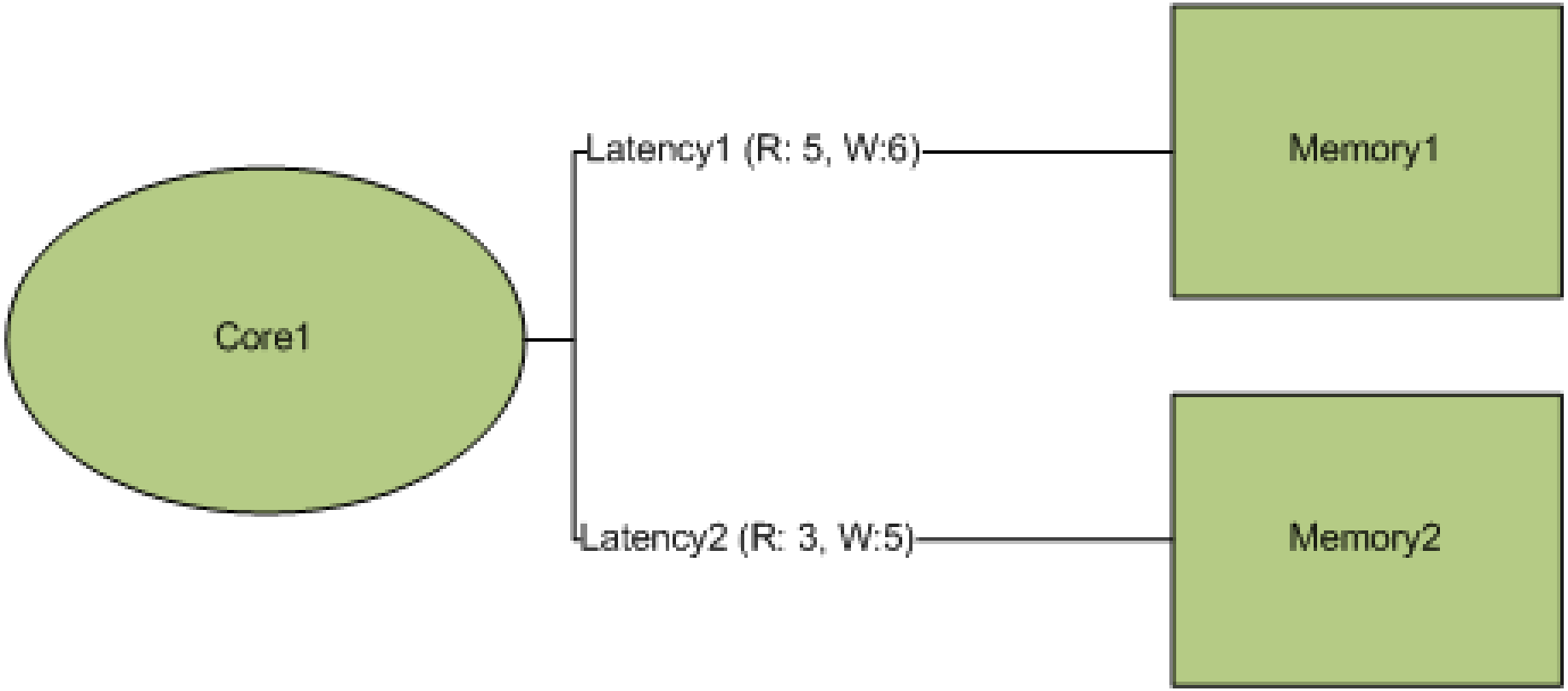
Attribute	Type	Mul.	Valid Values	Description
xAccessPattern	String	1	String	Timing Architects specific value. Will be used in the future to determine the access pattern of a memory element.
type	*MemoryType*	1	RAM/CACHE/FLASH_INT/FLASH_EXT	Specifies the type of this Memory.
size	Long	1	>0	Size of this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

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	*SchedType*	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

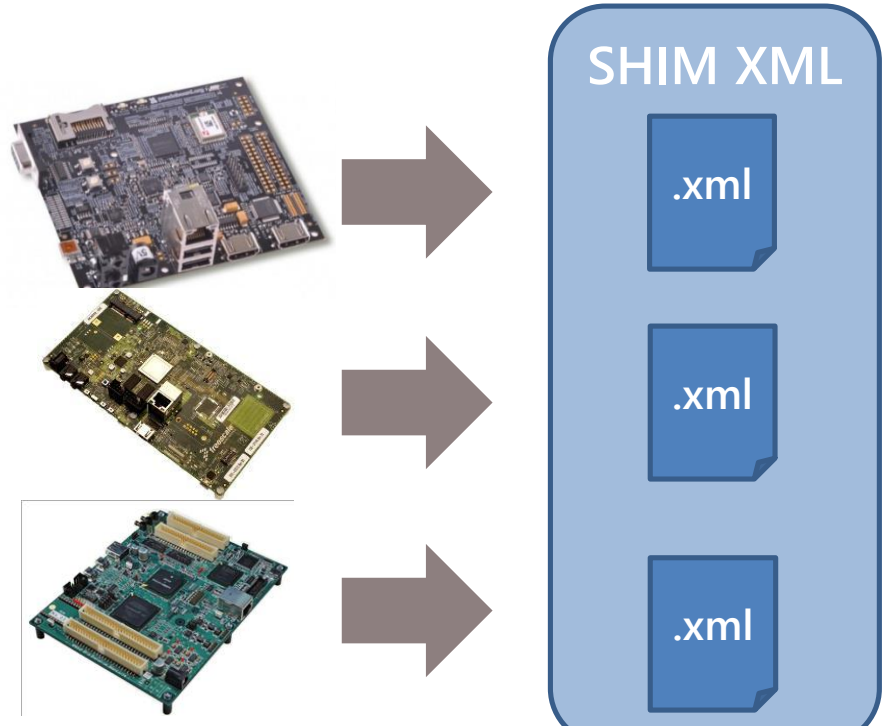
SHIM

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



1. Performance Estimation

Auto-parallelizing compiler

Parallelization tool

OS/middleware configurator

Performance analysis tool

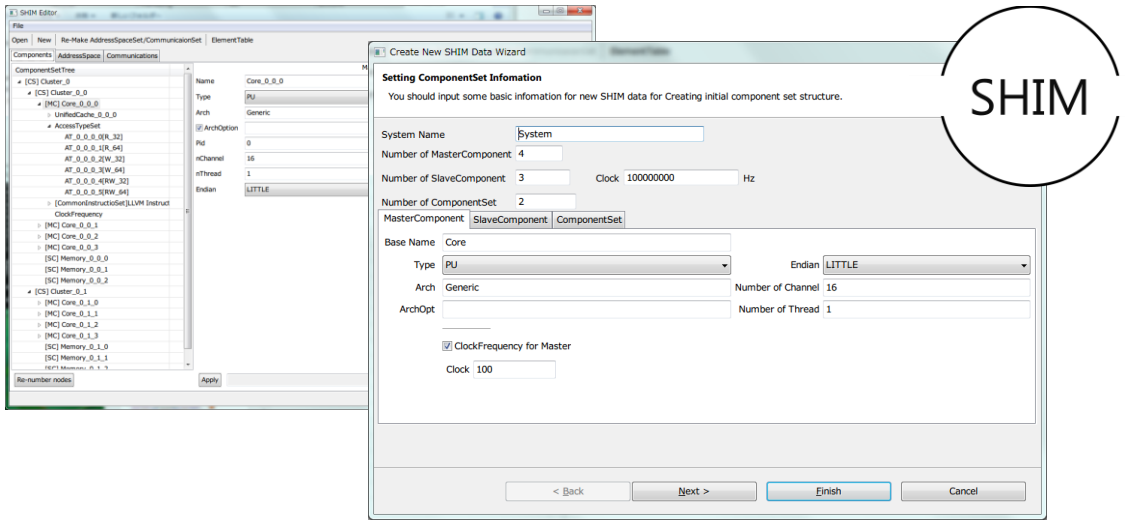
2. Tool Configuration

3. Hardware Modeling

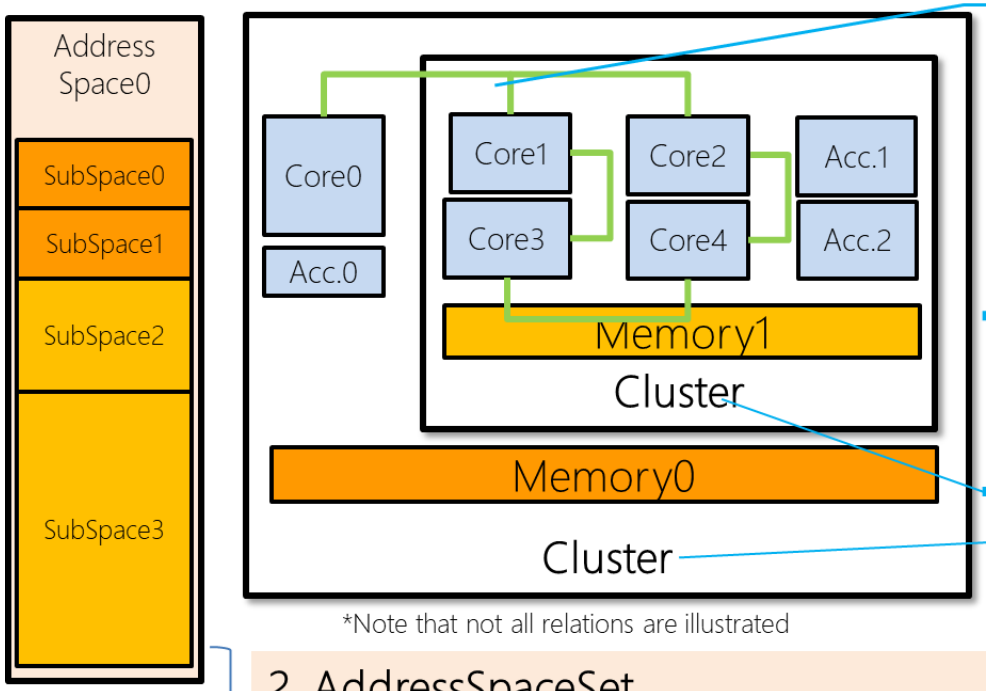
SHIM 1.0

- 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 XML schema



*Note that not all relations are illustrated

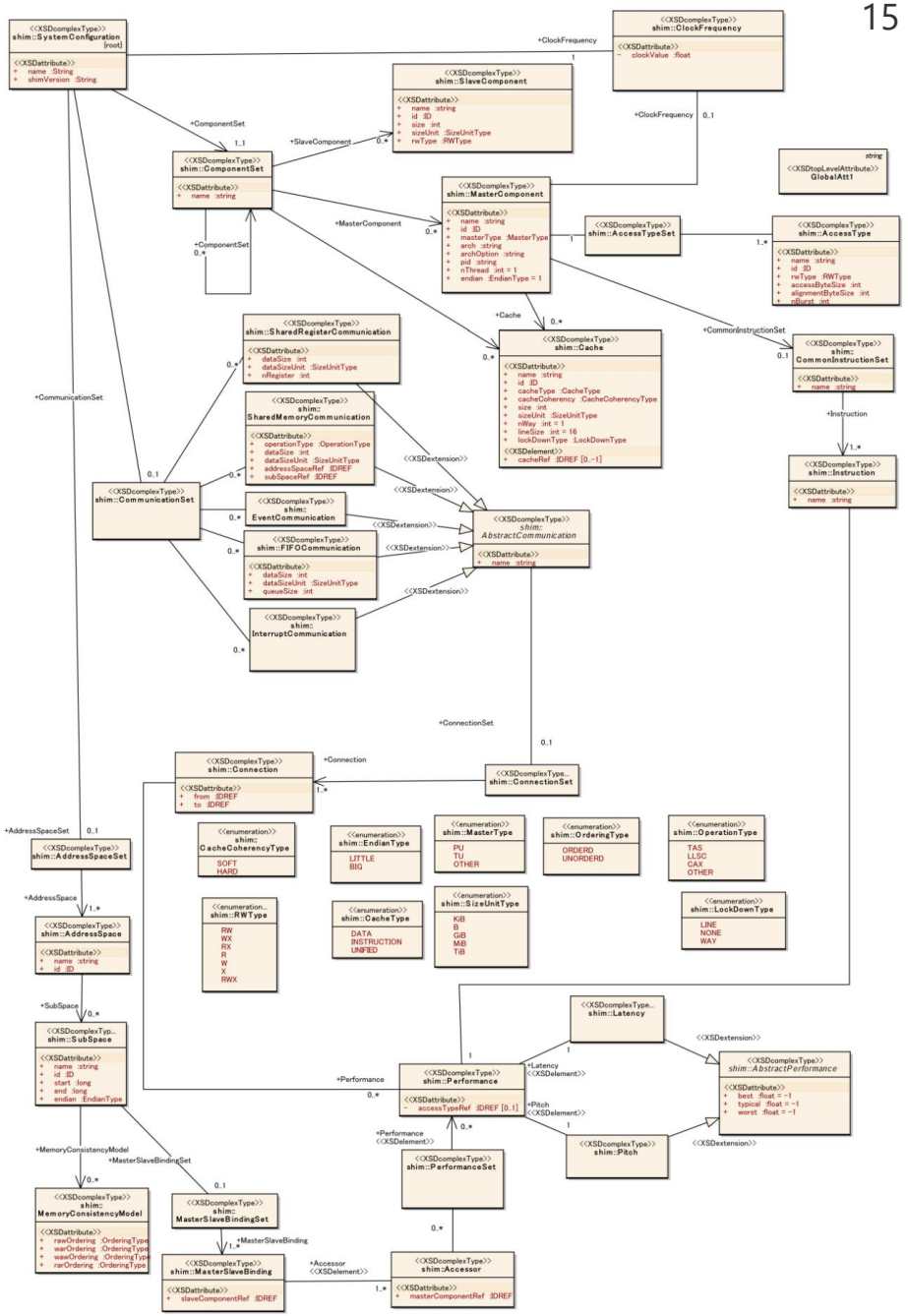
3. CommunicationSet
Inter-core comm., has PerformanceSet containing Latency and Pitch in cycles

MasterComponent:
Cache, ClockFrequency, CommonInstructionSet,AccessTypeSet

SlaveComponent: size, RWType

1. ComponentSet
Nest itself, can contain multiple, but single root

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



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ニーズ
  - 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**