

# MCoT? マルチコア・オブ・シングス?

「SHIM」「MBP」につづく「第三の活動軸」発足へ

2016-11  
ガイオテクノロジー(株) 岩井 陽二

# すみません..もうすこしの辛抱です



JASA/ETのホームページから画像引用

# この場でお話すること

- マルチコアオブシングスって？ 第三の活動軸って何？
- 現場を眺めると..いろいろ取り組めそうだな
- なるほど..難しそうだね
- ではどこから手をつけようか？
- WG（委員会）を立ち上げよう
- ゴール
- マイルストーン
- 是非 ご参加を（見学を）

# マルチコアオブシングスって？

- かつては産業界の米と呼ばれた半導体（ここではCPU/MPUを想像してください）
- でも..半導体自体は、ますます重要になっている  
自動車も電子化、ノウハウは半導体に集積
- そしてプロダクトはソフトウェアがインテリジェント化の鍵となる
- インテリジェントなソフトウェアは強力なコンピューティングパワーが必要
- そして..マルチコア・メニーコア
- インターネットだけでは語れない..マルチコア オブ シングス

## 第三の活動軸って何？

- EMC（組み込みマルチコアコンソーシアム）では、「SHIM」「MBP」と今後が期待されるアウトプットがある
- でも、まだやれることはありそう
- やるからには開発現場の役に立つモノを提供したい
- なにも欧米がアウトプットを出すのを待っていることもなかろう
- では、新任理事も第三の活動軸として、委員会（WG）を持ちましょう

# 現場を眺めると..いろいろ取り組みそうだ

## 現場（ソフトウェア側）で聞く 悩み/課題 の一例

- ツールの選び方
- ソフトウェアのデバッグ方法
- マルチコア活用のテクニックのメリット/デメリット（なぜその方法が良いのか?根拠説明で苦戦）
- ISO26を想定した場合の設計の考え方のメリット/デメリット（なぜその方法が良いのか?根拠説明で苦戦）
- ハードにばかり目が、意識が、向かってしまう
- SMP/AMP/BMPそれぞれの選定目安
- RTOSの選び方
- アプリケーションのシミュレータ構築方法（実機だけでは厳しい認識）
- コア間の実行タイミング評価
- サンプルアプリケーション
- アーキテクチャ設計指針の目安
- マルチコア移行時の機能分割方法
- マルチコア移行時のオートコード方法
- OEMとサプライヤの分業境界
- コア数や(AMP/SMPなどの)利用方針に影響されないプラットフォーム
- 結合評価方法/統合評価方法
- 性能評価方法/不具合検出方法
- ツールチェーン統合（設計－評価の統合）
- マルチコア（並列動作）対応設計書式の標準（設計書レビューの負担懸念）
- 負荷分散設計をする場合の指標（実装後の調整値として対応することになり、設計時に判断できないことへの懸念）
- 設計前の能力測定に時間を要する
- 並列化すべき方法論がない
- OSの使い方においてマルチコア特有の手法に関する情報が乏しい
- シングルコアで発生しない不具合やリスク（として何を想定しておくべきか）の目安？（目安から設計/検証の値ごろ感を掴みたい）
- 品確の目安

課題はたくさんありそう

マルチコア/メニーコアというキーワードは  
数年前から耳にするが、その間になぜ  
課題が解消されなかったのだろうか？

# なるほど..難しそうだね (1)

## 非常に複雑でいろいろなことを考えなければならない

- めざ・組込み、はどのような状況下でも決められた性能を維持して動作するのが当然、という既成意識
- 性能保証要求を満たすのは当たり前..となるが、その要求は難しい要求であることにまだ気づいていない人は多そう

## 組込みマルチコアプロセッサのソフトウェアプラットフォームや、HALのようなソフトウェア開発支援技術、あるいはマルチコアを意識させない開発環境、がないと..

- メモリ共有の有無やコアの対称性によって異なるプログラムを組む必要
- さまざまなタスクを、その負荷に応じて各コアに分散するところまで開発者が考えなければならない
- でも..上記の開発手法、開発環境が普及すると、全体を把握、理解できる、技術者は..

## にもかかわらず、マルチ/メニーコア化はどんどん進む..

- 近いうちにソフトウェア開発の難しさが進歩のボトルネックになるのは目に見えている

# なるほど..難しそうだね (2)

## 現場は思考停止状態？

- 「誰かが考えているだろう」的お見合い状態になっていませんか？
- ハードの熱問題はソフトウェアが解決してくれるだろう、と勝手に期待
- ソフトウェアの問題はハードがどうにかしてくれるだろう、と勝手に期待
- マルチ/メニーコアを使う（≠使い切る）ために、従来の開発体制やメソッドを見直す必要があると気づいていても行動できていないのでは？

## 競争相手を意識すると..うかつなことはいえない？

- 競争領域/非競争領域（協業領域）
- 誰が線引きするの？

## 地政学的？宿命

- EUでは国家横断プロジェクトが横断的、継続的に形成され、1企業ではなし得ないアウトプットをどんどん生み出している
- 日本は？日本だけ？

# ではどこから手をつけようか？ (1)

- ツールの選び方？ RTOSの選び方？
- ツールチェーン（設計-評価の統合）？プラットフォーム？
- マルチコア活用方法と選択（根拠説明）？
- SMP/AMP/BMPそれぞれの選択目安？
- サンプルアプリケーション？アーキテクチャ設計指針？
- マルチコア移行時の機能分割方法？
- マルチコア移行時のオートコード方法？
- マルチコア移行時の設計指針？
- 設計前の能力測定？
- 性能評価方法？不具合検出方法？
- 品質確認の指針？
- ビジネスモデル？
- 海外動向？海外プロジェクトの研究？

(利用者目線での) 非競争領域、技術トレンドの集約と配信、実務に転用可能な情報・データ作成と配信が良いだろう..と思う

# ではどこから手をつけようか？ (2)

- 医療？ロボット？自動車？OA？

自動車から始め（その後に）他へ横展開

- 議論の出発点は？

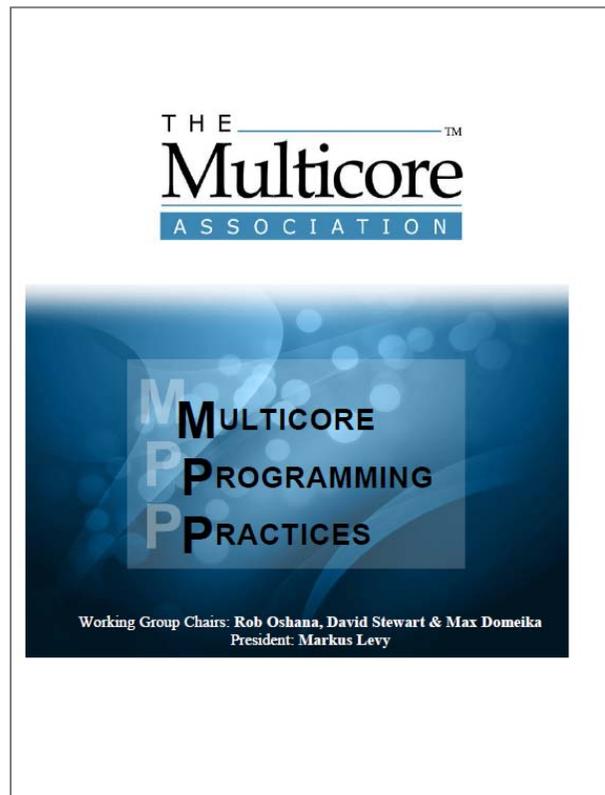
MCA\*のMPP本を議論の出発点に  
\*MultiCore Association

- メンバーは？

EMCメンバーの参加者と  
自動車に関連する仕事に接点のある方

# MPP\*本

- MCAのMPP\*本 \*Multicore Programming Practices
- マルチコアに関するの議論の出発点（きっかけ）として適当と考えた



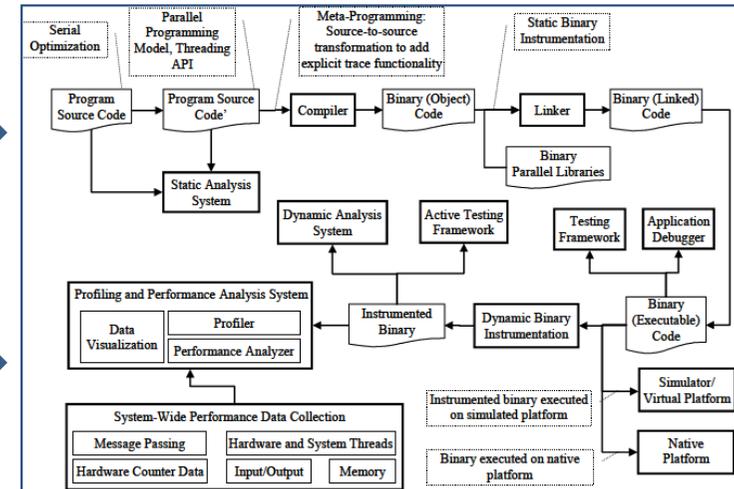
The image shows the Table of Contents page from the book. The page is titled 'Table of Contents' and lists the chapters and their page numbers. The chapters are: Chapter 1: Introduction & Business Overview, Chapter 2: Overview of Available Technology, and Chapter 3: Analysis and High Level Design. The page number 'Page 2' is visible at the bottom right.

Table of Contents	
Forward: Motivation for this Multicore Programming Practices Guide	
Chapter 1: Introduction & Business Overview	
1.1 Introduction	10
1.2 Goal of MPP	11
1.3 Detailed Description	11
1.4 Business Impact	12
1.5 Target Audience	12
1.6 Applying MPP	13
1.7 Areas Outside the Scope of MPP	14
Chapter 2: Overview of Available Technology	
2.1 Introduction	16
2.2 Programming Languages	16
2.3 Implementing Parallelism: Programming Models/APIs	16
2.4 Multicore Architectures	16
2.5 Programming Tools	17
Chapter 3: Analysis and High Level Design	
3.1 Introduction	19
3.2 Analysis	19
3.3 Improving Serial Performance	19
3.3.1 Prepare	20
3.3.2 Measure	21
3.3.3 Tune	22
3.3.4 Assess	23
3.4 Understand the Application	24
3.4.1 Setting Speed-up Expectations	24
3.4.2 Task or Data Parallel Decomposition	25
3.4.3 Dependencies, Ordering, and Granularity	26
3.5 High-Level Design	28
3.5.1 Task Parallel Decomposition	28
3.5.2 Data Parallel Decomposition	29
3.5.3 Pipelined Decomposition	29
3.5.4 SIMD Processing	30
3.5.5 Data Dependencies	30
3.6 Communication and Synchronization	32
3.6.1 Shared Memory	32
3.6.2 Distributed Memory	33
3.7 Load Balancing	34

# MPP本

## • たとえば..「ツールの選び方」なら..

- システム・アーキテクチャ設計
  - 本資料のP.5[MPP本での参考文献一覧]が参考になる
- 機能・モジュール・詳細設計
  - 6.16 Using Parallel Libraries
- 単体試験
  - C. 1 Introduction to Parallel Tool Categories
- 静的解析
  - 5.4 Static Code Analysis
- 結合試験
  - C. 1 Introduction to Parallel Tool Categories
  - 5.8.2 Logging
  - 5.8.3 Synchronization Points
  - 5.8.5 Simulation Techniques Summary
  - 5.8.6 Stress Tesing
- 実機評価または同等環境での評価
  - 5.5 Dynamic Code Analysis
- 構成管理
  - 該当なし



Multicore Programming Practicesから引用

# MPP本

- **たとえば..「ツールチェーン統合（設計－評価の統合）」なら..**
  - ツール一覧（ツールマッピング）
    - まとまっていない ※ところどころにオープンソースを紹介しているがリストにできるほどには紹介していない
  - ツール関連系
    - C. 1 Introduction to Parallel Tool Categories
- **たとえば..「性能評価方法/不具合検出方法」なら..**
  - かしこい（わかりやすい、つかいやすい、マルチコアに適した）プロファイラ
    - 6.6 Improving Load balancing(直接は関係ないが性能問題の時、考慮すべきこと)
  - コア間の実行タイミング評価（Vの左側/右側の両方）
    - 5.5 Dynamic Code Analysis
  - ソフトウェアのデバッグ方法（Vの右側）
    - 5.3 Debug Tool Support
    - 5.8.2 Logging
    - 5.8.3 Synchronization Points
    - 6.3 Using Compiler Flags（デバッグの準備）
  - アプリケーションのシミュレータ構築方法（実機だけでは厳しいのでは？）
    - 該当なし
    - 「5.8.5 Simulation Techniques Summary」から始めるとおかしくなるかも？（ユースケースによる）

# MPP本

## • 参考（MPPを読んでみて）

- かなりページを割いていた、第3章、第4章について
- 「(D) マルチコア移行時の機能分割方法」
  - 第4章は、上記の実際に用いる具体的な手段の説明が主
  - MCAPI/MRAPIにつながる説明がメインに展開されています
  - 第3章は第4章の前提条件となるマルチコアの構造的な基礎知識
    - 3.5: 上流設計で考慮される事柄
      - タスク並列化
      - データ並列化
      - Pipeラインの概念

### MCA IMPLEMENTATIONS

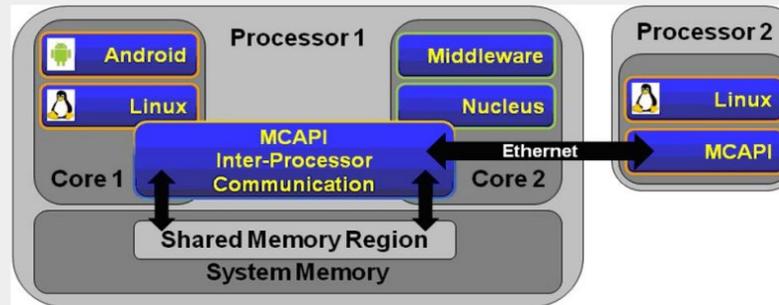
▼ Mentor Embedded MCAPI, a product of Mentor Graphics, MCA Board Member

Mentor Embedded MCAPI™ provides fast IPC messaging and application code portability with abstraction layers for operating system and physical transport. Android/Linux and the Nucleus™ Real-Time Operating System (OS) are fully supported while Mentor Services can assist with porting to other operating systems. <http://www.mentor.com/embedded-software/events/esc/upload/multicore-esc-ds.pdf>

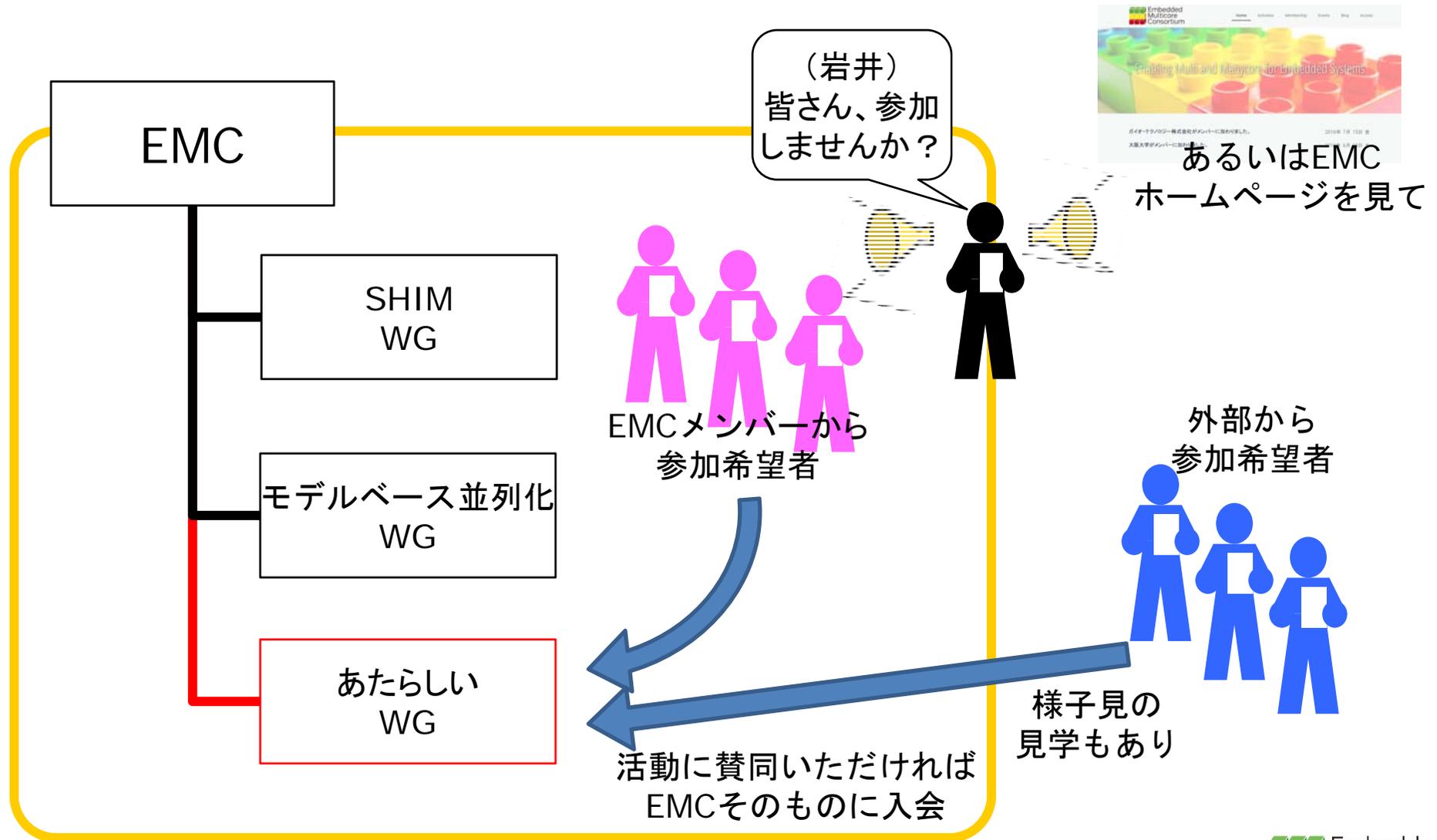
MCA Component used in Product:  
MCAPI

Product Benefit:  
With an expanding adoption of MCAPI™, customers will have a wider array of tools and system software available to them. From the vendor's perspective, a standard like MCAPI™ increases the partnership opportunities and prospective customer base.

Client Benefit:  
MCAPI™ provide a fast, lightweight, and scalable IPC solution that is highly portable to new target, transport, and OS - adopting the MCA standard insures application portability to new hardware and operating systems, allowing system designers to maximize code reuse. MCAPI™ provides a simple and intuitive API allowing application developers to create endpoints, similar to TCP/UDP sockets, for sending and receiving data using block and non-blocking calls. MCAPI™ supports multiple communication links with prioritization and Mentor Embedded MCAPI™ supports prioritization of multiple message and channels.



# WG（委員会）を立ち上げよう



# マルチコア活用（実用）委員会（仮称）

- WG構成
  - 委員長：岩井（ガイオ・テクノロジー）
- 活動計画
  - 定例委員会を開催  
（+立ち上げ期は中核になるメンバーの集中討議も？）
- 期間：2017/1～2018/3（原則として継続）  
2016/11～2016/12は準備活動期間
- 対象：マルチコアを積極的に活用する方法やマテリアル
- 活動内容
  - 詳細ガイドの作成・配布
  - 手順書やテンプレートの作成・配布
  - ツール要件（集）の検討および情報配信
  - 実証PJやツール試作および提供

# ゴール（候補）

- マルチコア前提の開発が直面する課題を明確にする
  - 課題リストを毎年アップデートし開示する
  - 自動車分野から他の分野に展開する
- 課題のうち、解決できるテーマに対しアウトプットを示す
  - マイルストーンにもとづきアウトプット（成果物）をリリース
- アウトプットを開示し 広く利用を促進する
  - アウトプットは国内外に開示し 広く利用を促進する

# アウトプット（候補）



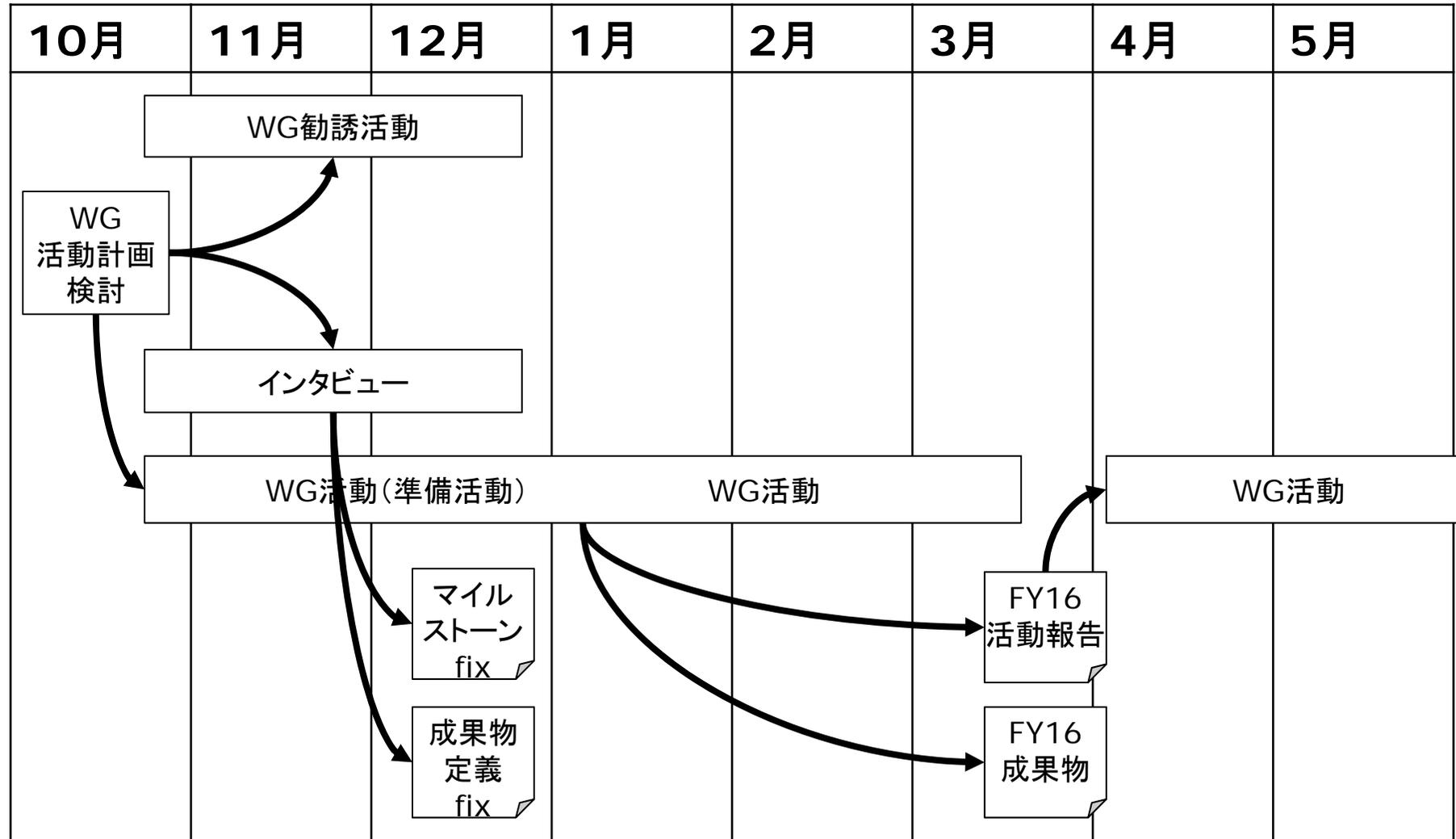
パッケージをもとにMPPの付属書を作ってみる？

上記3点はパッケージとして整備も良いかと  
たとえば..マルチコア活用(実用)パッケージ

# マイルストーン（素案）

FY16	FY17	FY18	FY19
WG発足	活動をひろく認知いただく	実績を出す (前年成果物で効果)	日本を代表する活動体
マイルストーン決定	マルチコア活用(実用)パッケージ V1	マルチコア活用(実用)パッケージ V2	マルチコア活用(実用)パッケージ V3
	海外動向レポート FY17	海外動向レポート FY18	海外動向レポート FY19
	技術動向レポート FY17	技術動向レポート FY18	技術動向レポート FY19
		実証PJ or 試作ツール(FY18)	実証PJ or 試作ツール(FY19)
		MPP付書発行	

# 当面の段取り



# 必要とあれば他の活動体との連携も

- 国内でマルチコアをテーマに扱う活動体
- MCA
- AMALTHEAをはじめとした欧州プロジェクト

...ほか 連携が市場に有益と判断できる活動体すべてにオープン

# 是非WGにご参加（もしくは見学）を

入会フォームがあるわけではないです  
下記のテキスト程度のメッセージを  
添えてもらうと良いかもしれません

申し込み、お問い合わせは、

EMCのホームページの「ACCESS」  
お問い合わせフォーム

か

わたくし、岩井（iwai@gaio.co.jp）

あてにお願いします

EMC マルチコア活用（実用）委員会 あて

- > WG活動に参加します
- > WG活動を見学したいです

会社名：□□□□□□□□

部署名：□□□□□□□□

ご連絡先（電子メール）：xxxxx@xxxx.xx.xx

メッセージ

（参加にあたりなにか伝えておきたいことが  
あれば書き添えてください）

# あ、そうそう

この後（ETフェスタのことです）  
EMCのブースにも遊びに来てください

JASA特設パビリオン内 [展示番号 D-15-20](#) です



お問い合わせは  
[www.embeddedmulticore.org](http://www.embeddedmulticore.org)