



Industrial experiences on using MetaEdit+ in embedded software development

MetaEdit+
組込みソフトウェア開発の顧客事例



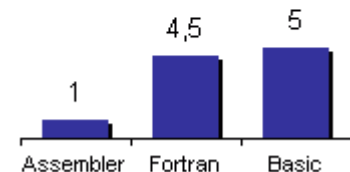
内容

- Domain-Specific Modeling
 - イントロダクション
 - 各種事例
- 産業界からの事例報告
 - スポーツコンピュータ
 - ホームオートメーション
 - 防衛向け通信機器
- まとめ



これ以上の生産性向上は汎用言語では見込めない

- “ソフトウェアエンジニアリングの歴史は抽象度の向上と共にあった”
- 新しい汎用プログラミング言語でも生産性は上がらない
- しかし開発における抽象度はまだ上げることが出来る、、
- 僅かな成果に我慢することや自由度を犠牲にすることなく...



*Software Productivity Research & Capers Jones, 2002



産業界の事例



“2週間かかっていたモジュールの開発（設計開始から製品化まで）は1日で済んでいる”



“自動生成されるコードは明らかに品質が良くなった。モデリング言語によってエラーを排除できる仕組みが取れるので”

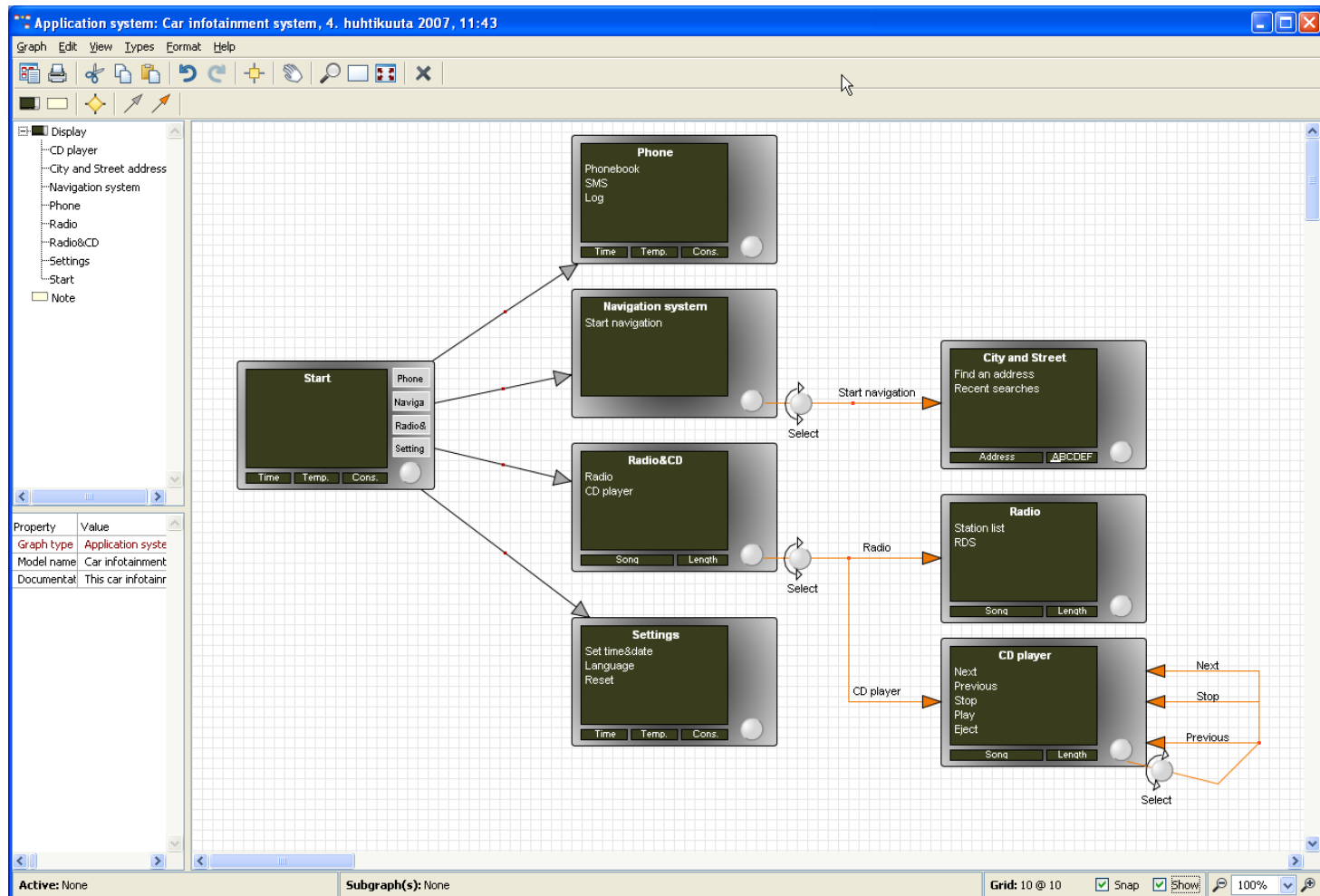


“DSMによって開発は飛躍的に加速化され、また容易になった。旧来の人手に頼ったコード実装に比較して”

* source: www.metacase.com/cases/

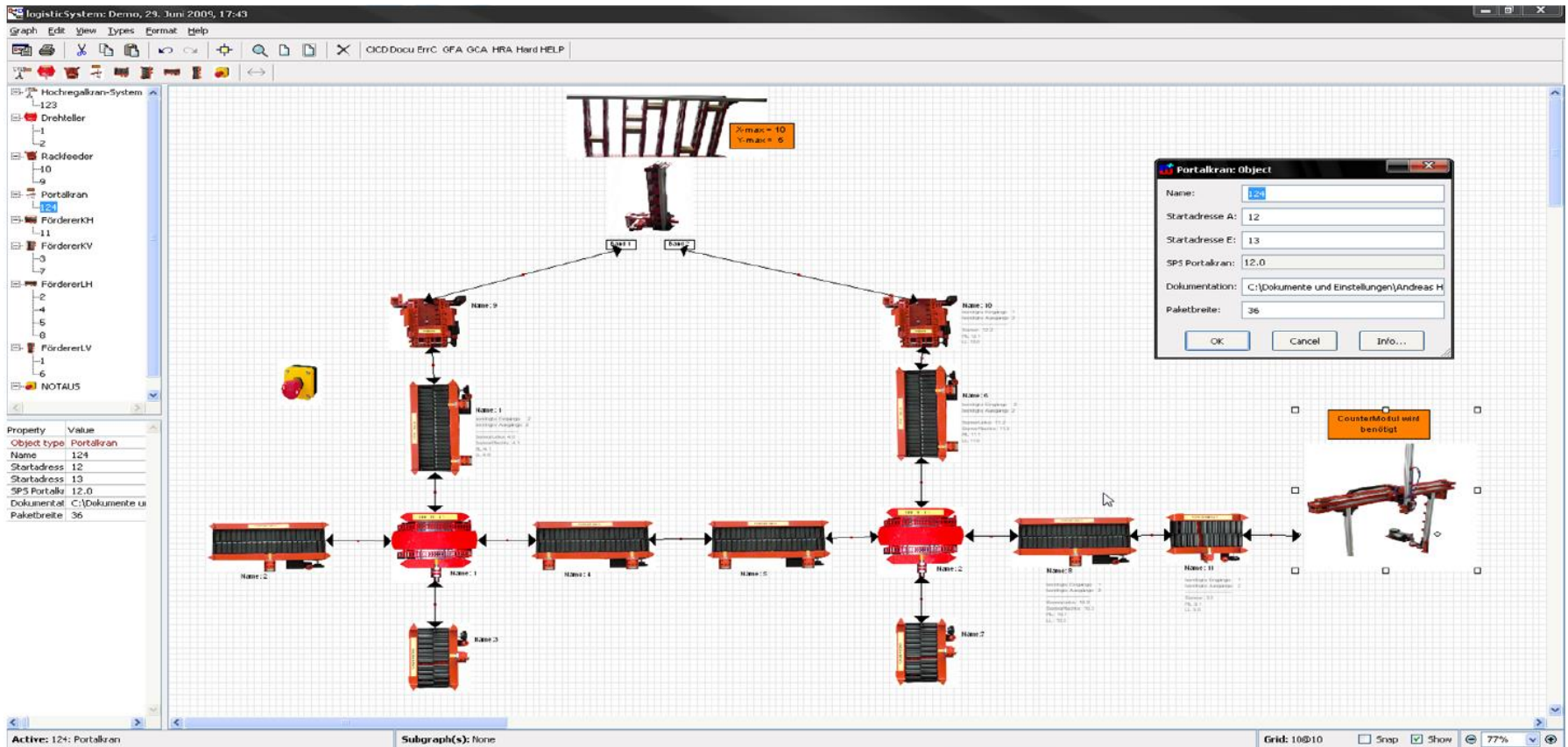


自動車のヒューマン・マシン・インターフェイス (カーナビ)



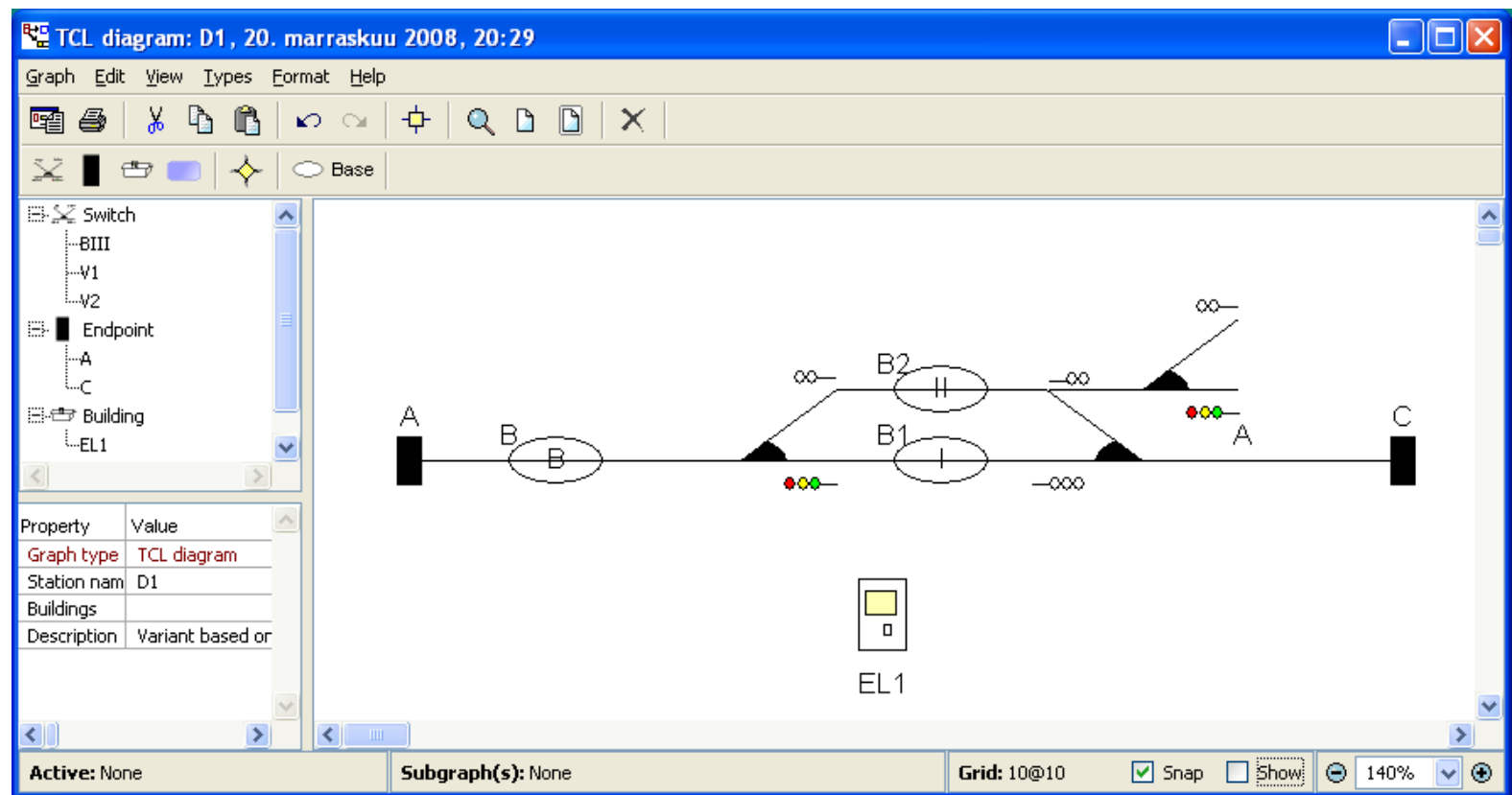


産業機器 (FA)



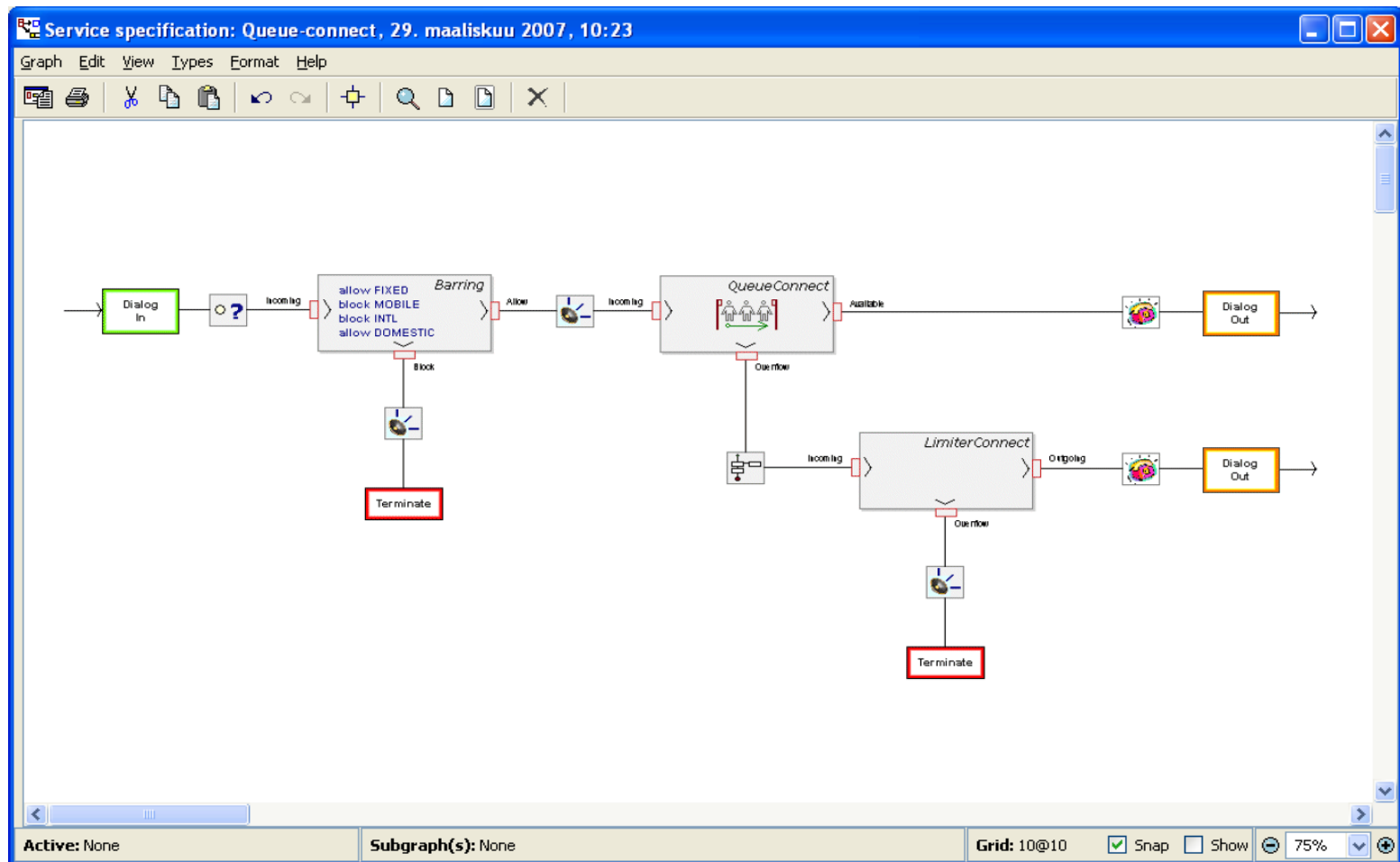


列車の連結制御システム



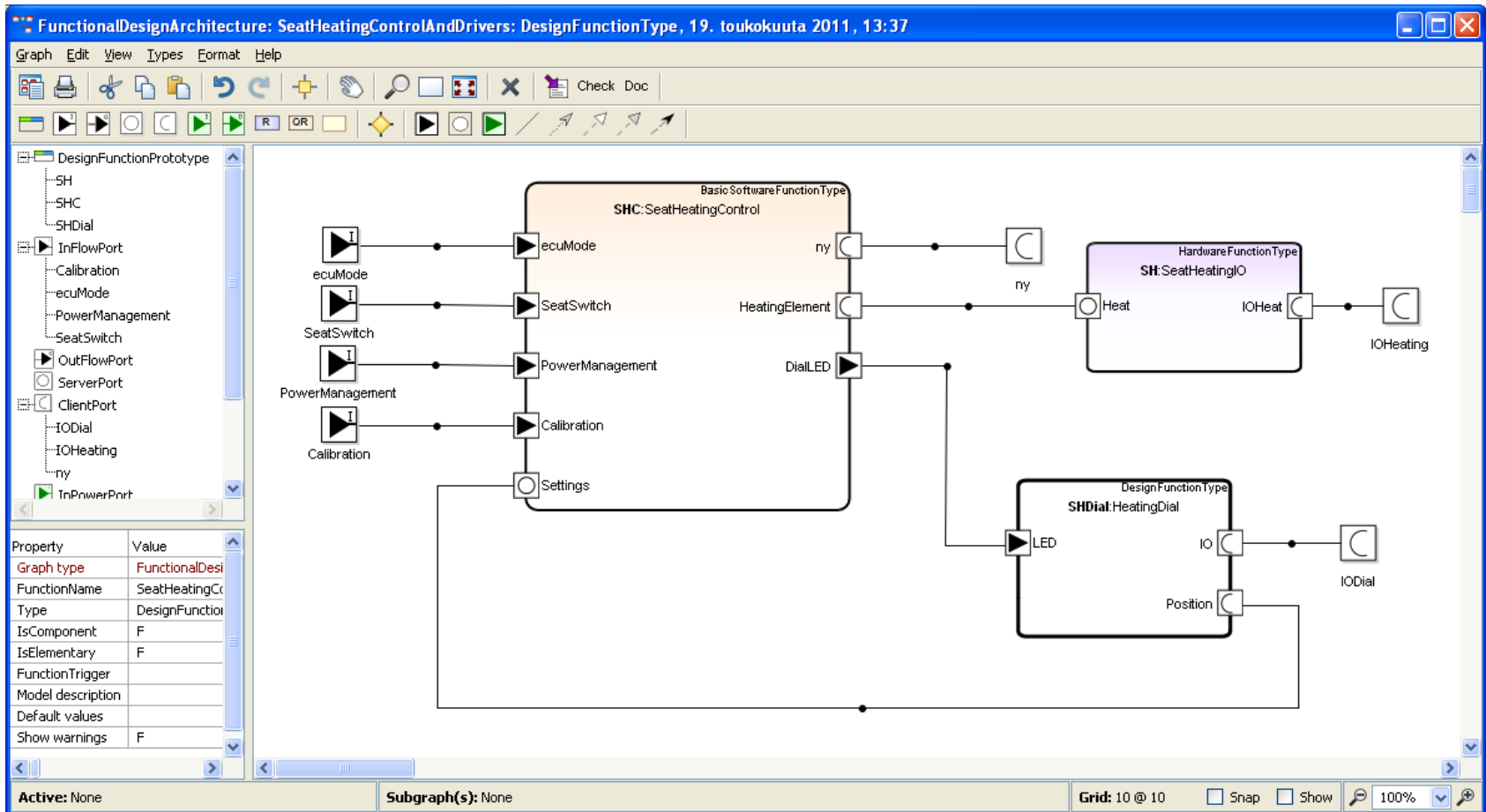


電話通信サービス (IMS Service Creation)



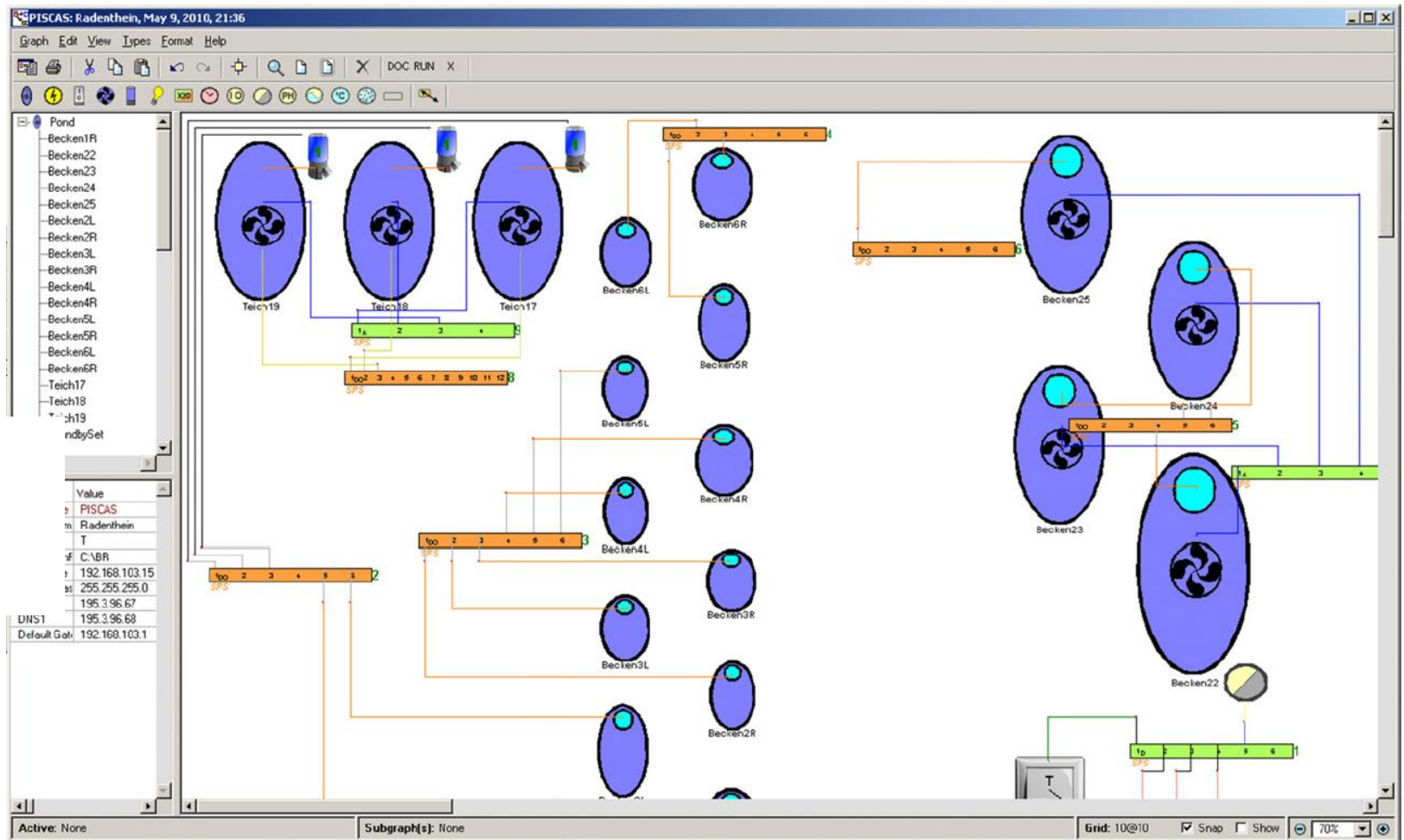


車載向けアーキテクチャモデリング (AUTOSAR、EAST-ADLなど)





養魚場の管理制御システム





3つの事例の詳細

- スポーツコンピュータ (Polar Electro 社)

<http://www.dsmforum.org/events/DSM09/Papers/Karna.pdf>

- ホームオートメーション (Ouman 社)

http://www.codegeneration.net/cg2011/slides/Puolitaival_CG2011.pdf

- 防衛向け通信機器 (Electrobit 社)

http://www.thinkmind.org/download.php?articleid=valid_2011_5_20_40049



Case 1: Polar社のスポーツコンピュータ



- 搭載される各フィーチャーは:
- Heart rate measuring, analysis and visualization
- Calorie calculation, like current, cumulative, expenditure rate, active time
- Speed: current, average, maximum
- Distance, based on interval, trip, recovery
- Altimeter, vertical speed, altitude alarms, slope counter, graphical trend
- Cycling information like pedaling rate and cycling power
- Barometer, pressure drop alarm, graphical trend
- Exercise diaries
- Sensor connectivity (heart rate, speed, cadence, power, GPS)
- Compass, Temperature, Odometer, Logbooks, etc.



About the product line (Kärnä et al. 2009)

- Polar 社はUIのアプリ開発にフォーカスした
 - 最も大きいアプリケーションソフトで
 - 開発期間の40-50% を要していた
 - 製品間で大きく異なる
 - Results of domain analysis showed
 - 70% のUIアプリケーションコードは容易に自動生成できる
 - 25% も多分自動化できる
 - 5% のUI の特別なコードはDSMの自動生成対象には向かない
- ソフトウェア開発は限られたリソースの制約内で:
 - メモリ、プロセッサの速度、バッテリーの寿命

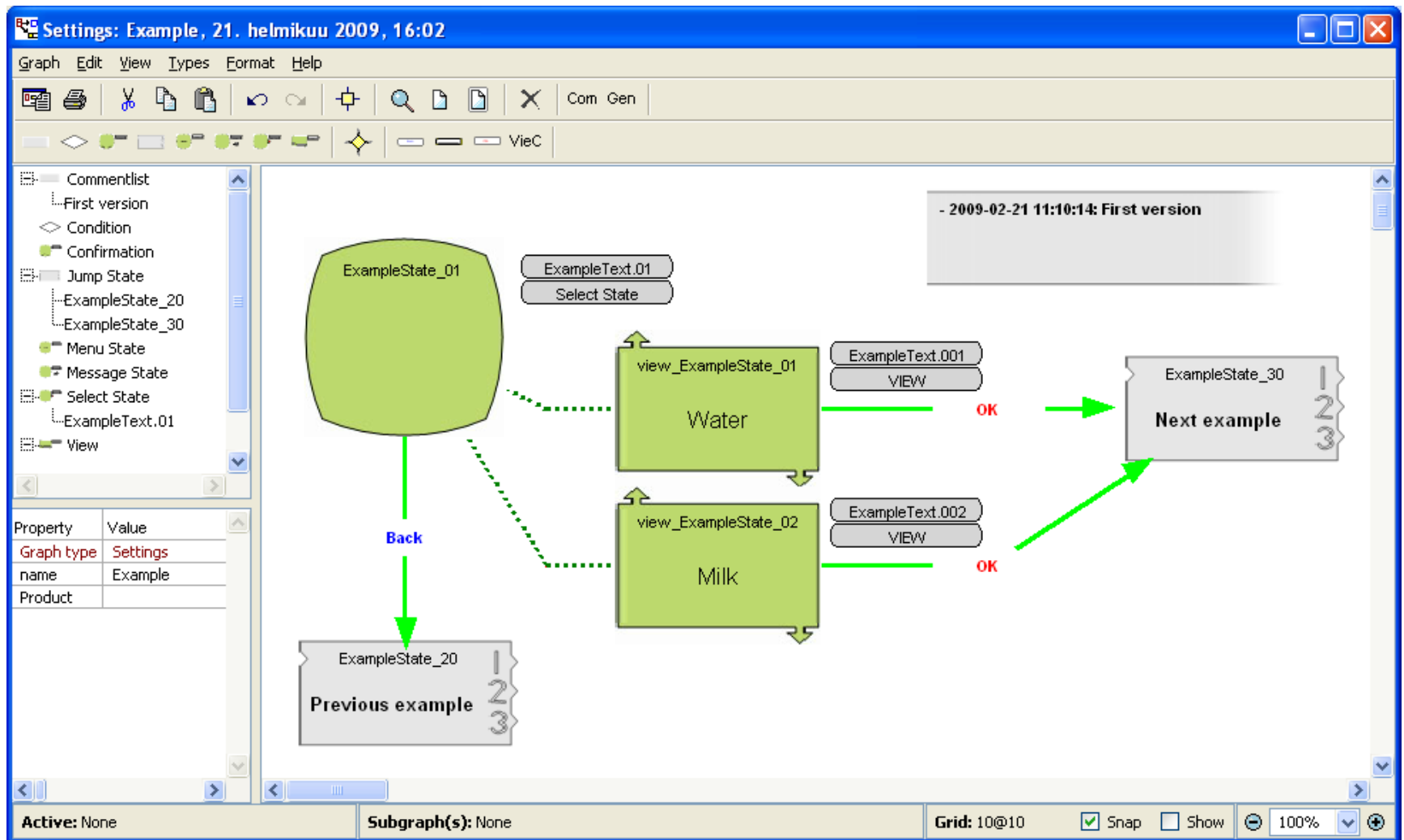


DSM に求めた要件

- Polar 社で（コンサルなど外部の手伝いなしに）モデリング言語とジェネレータを作る
- DSM 環境に求めた主な要件：
 - UIアプリ開発の生産性向上
 - コード品質とメンテナンス性の向上
 - 人手によって仕様からコードにデータをコピーすることを削減
 - 新しく入った開発者への導入障壁を排除
 - 新人でも熟練者でも使用しやすいこと
- これら要件がどのように満たされるかを評価した
 - デザインから開始してアプリケーションが準備できるまで



UI アプリケーションのデザイン例





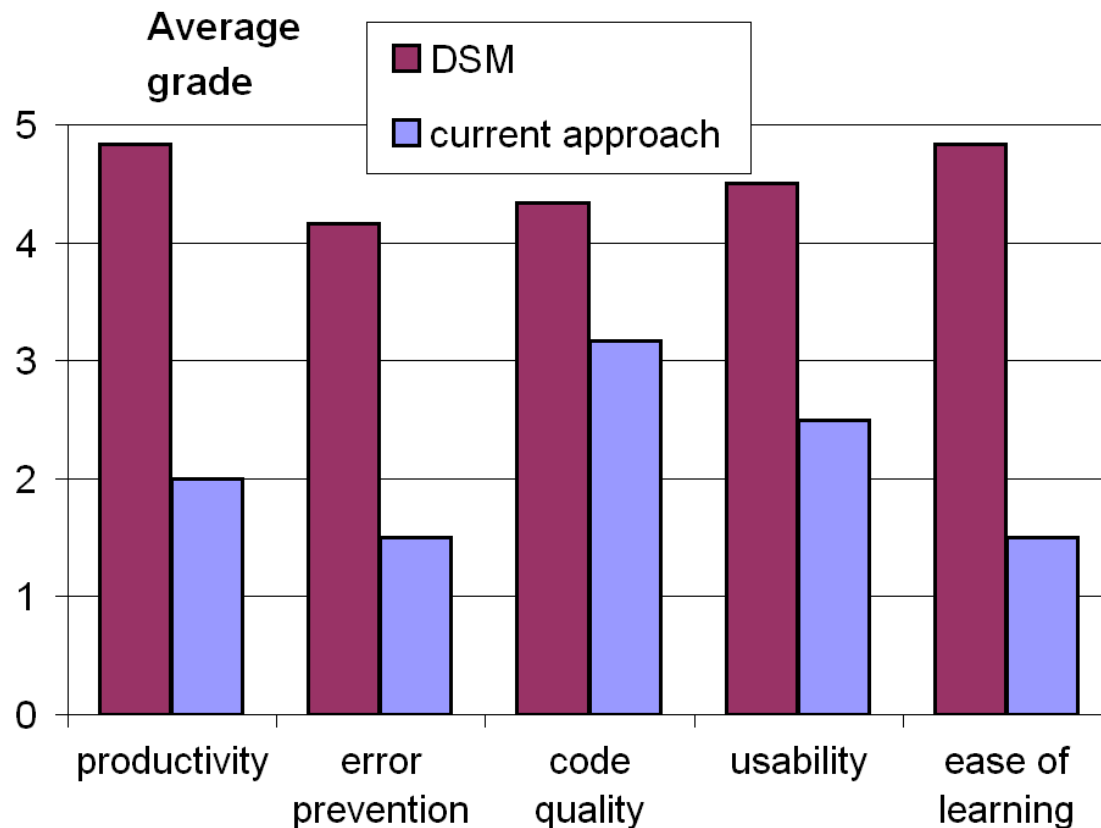
評価方法

- 評価にかかるコストと、再現可能で満足いく結果とのバランス
- 2つの評価・研究手段
 - － 研究実験
 - 小さくて、代表的なフィーチャを実装
 - 6人の現開発者が別々に独立して、それぞれ実装（計6つ）
 - － パイロットプロジェクト
 - ある製品全体の大きな部分を実装（64%）
 - 1人
- DSMの活用と従来手法を比較



開発者の意見・感想

- 6 人の開発者による研究実験：各担当者の感想
 - － 2時間の調査で結果を得た (5段階評価, 5 がベスト):

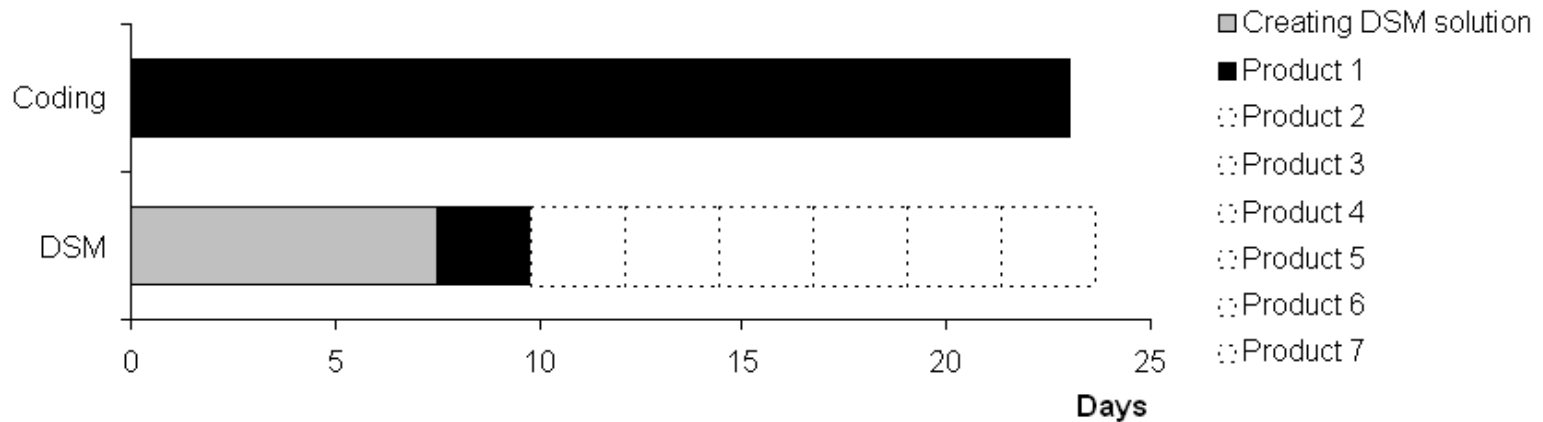




投資対効果: Polar社

■ パイロットプロジェクト

- ある製品全体の大きな部分を実装(64%)



■ DSM の開発は 60時間

■ Product development with:

- 従来手法: 23 日
- DSM: 2~3 日



Case 2: ホームオートメーション (Puolitaival, 2011)





ホームオートメーション: 携帯電話からリモートコントロール

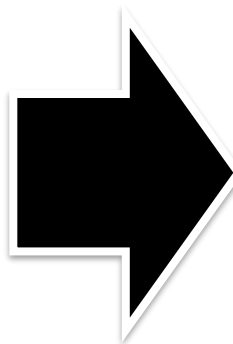
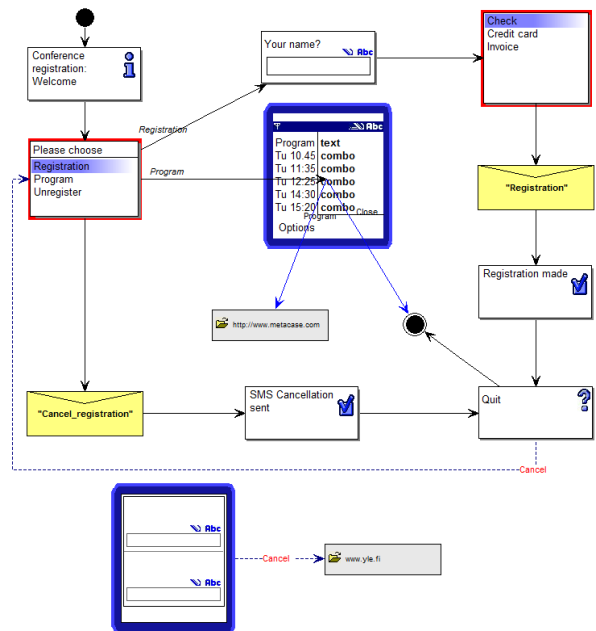
- Ouman manufactures home automation systems 社
- 様々な暖房装置の温度制御システムを製品として提供
- SMS を介したリモートコントロール



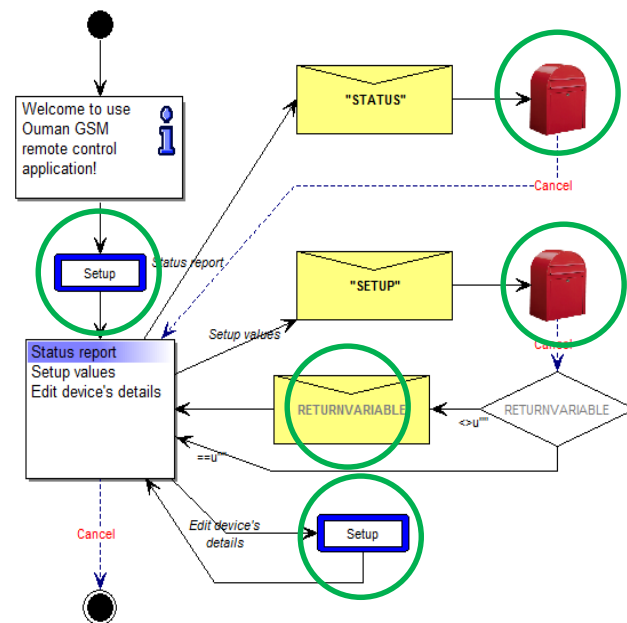


リモートコントロールのアプリを定義する言語

既存の携帯電話用の DSL

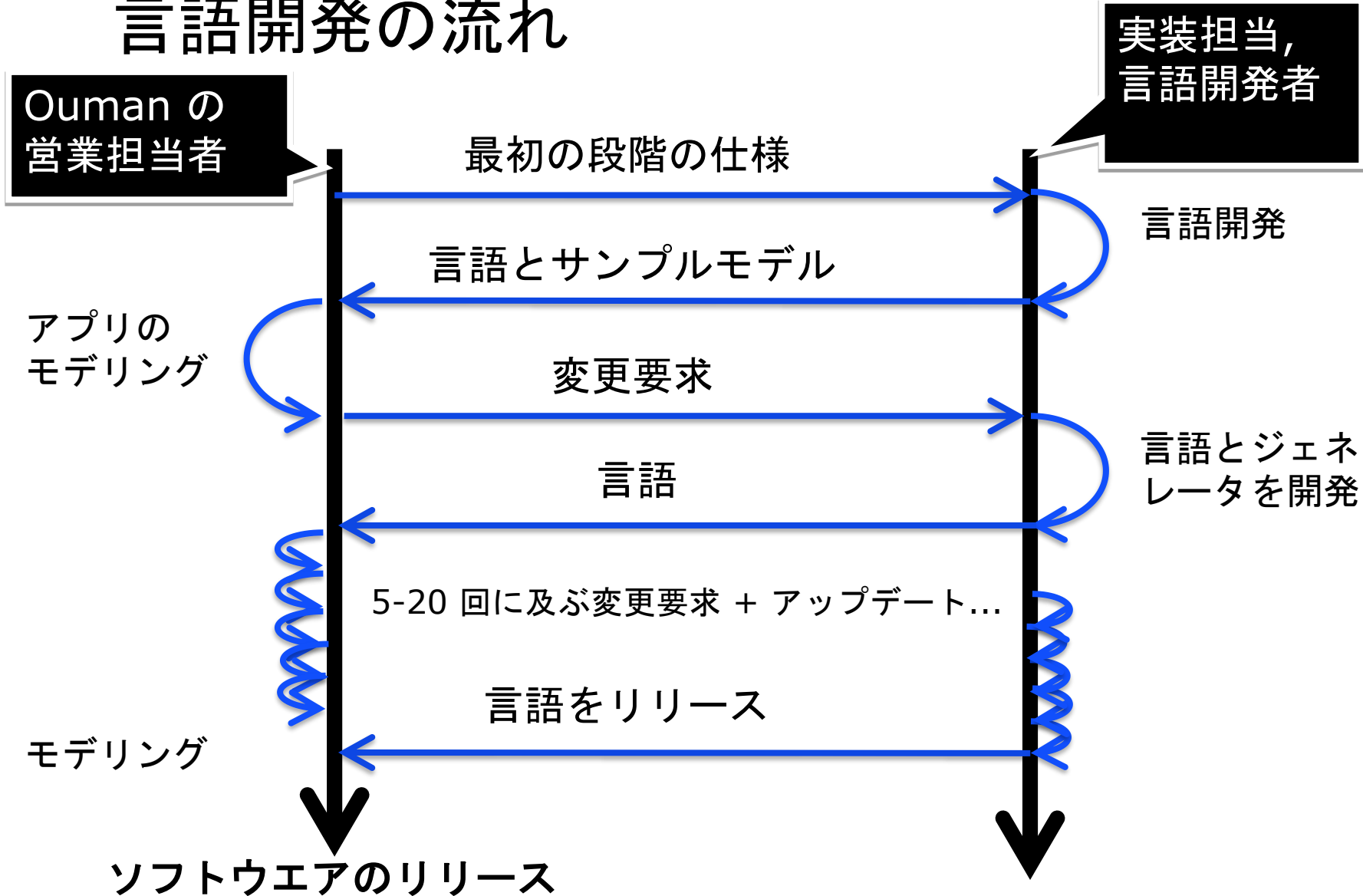


"Ouman社" の言語

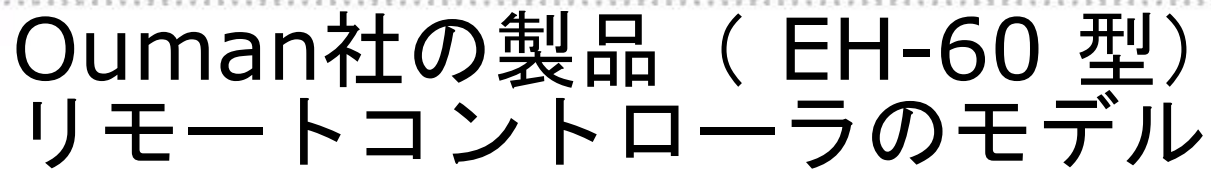




言語開発の流れ



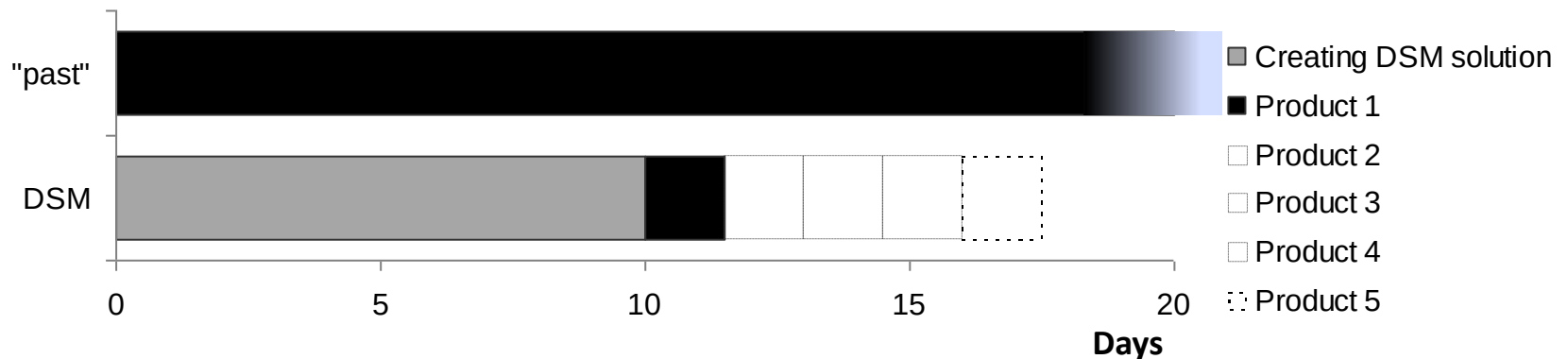
ソフトウェアのリリース





経験・成果

- DSM の開発期間: 2 週間
- 製品開発: 1~2 日
 - 従来手法との比較は無い。アウトソースしていたため
 - しかし開発コストは6分の1になった（ユーロで）



- DSM によってプログラマでなくともアプリを開発できるようになった
 - Ouman 社のマーケティング担当者がモデルを描いた



Case 3: 防衛向けの通信機器 (Puolitaival et al., 2011)





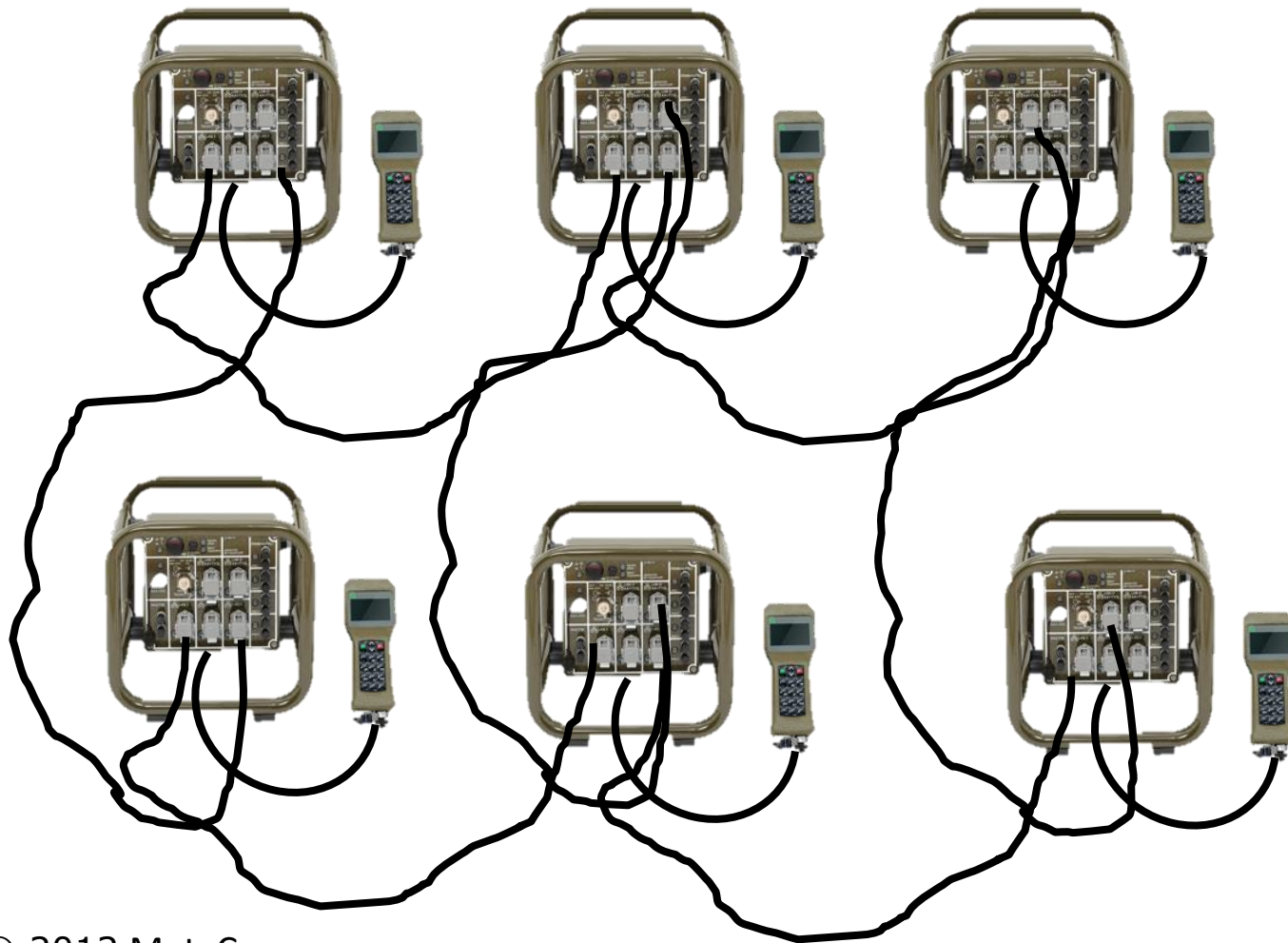
Elektrobit 社 防衛向け VoIP



- 堅牢な有線のVoIP電話でUDP/IPネットワークを使用
- Elektrobit 社
- 主な機能:
 - 構成の設定が容易
 - ポイント to ポイントコール
 - 全コール（呼び出し）
 - 軍用機器
 - とことんシンプルな設計



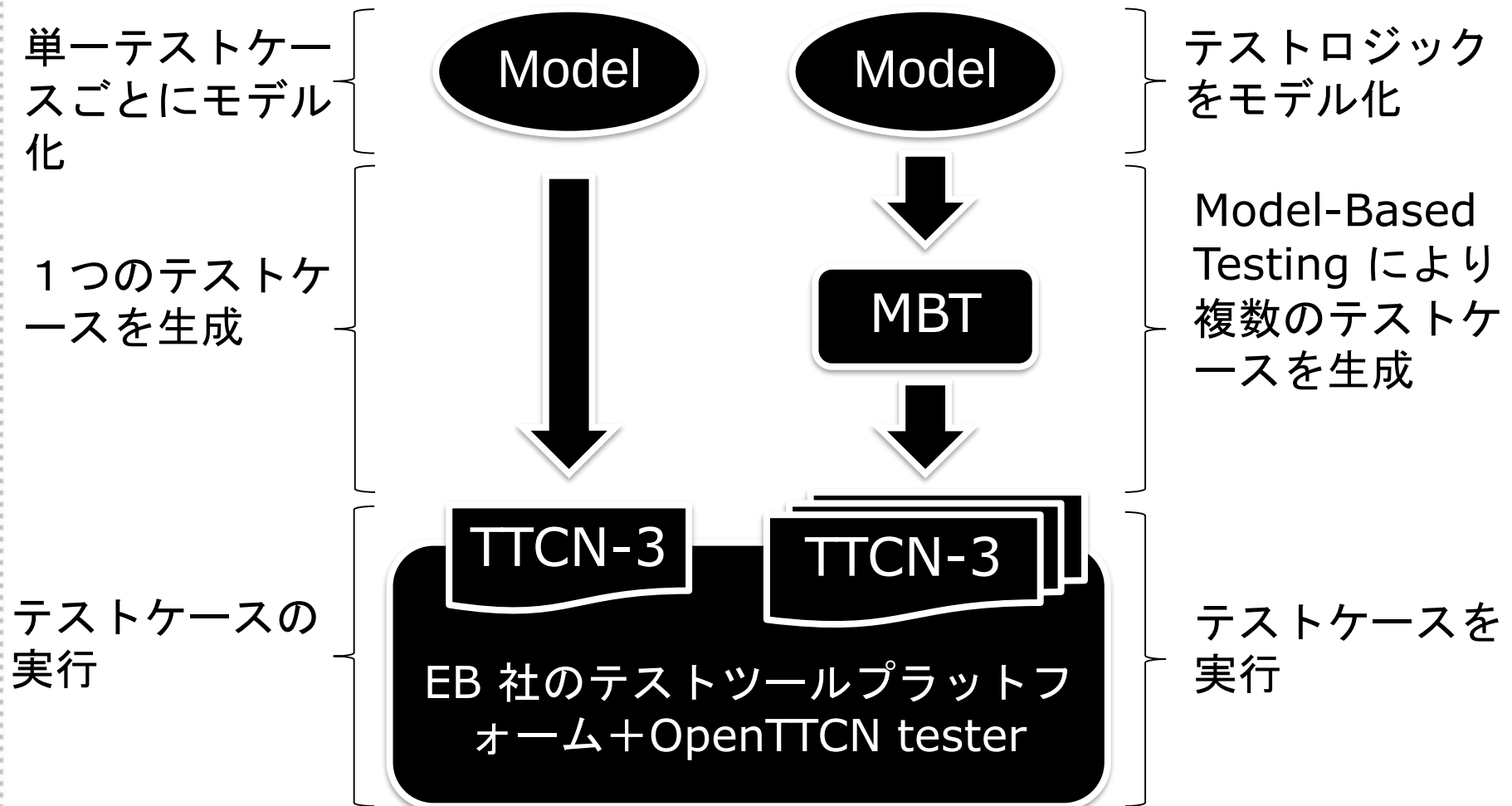
テストで直面する課題



ETC...



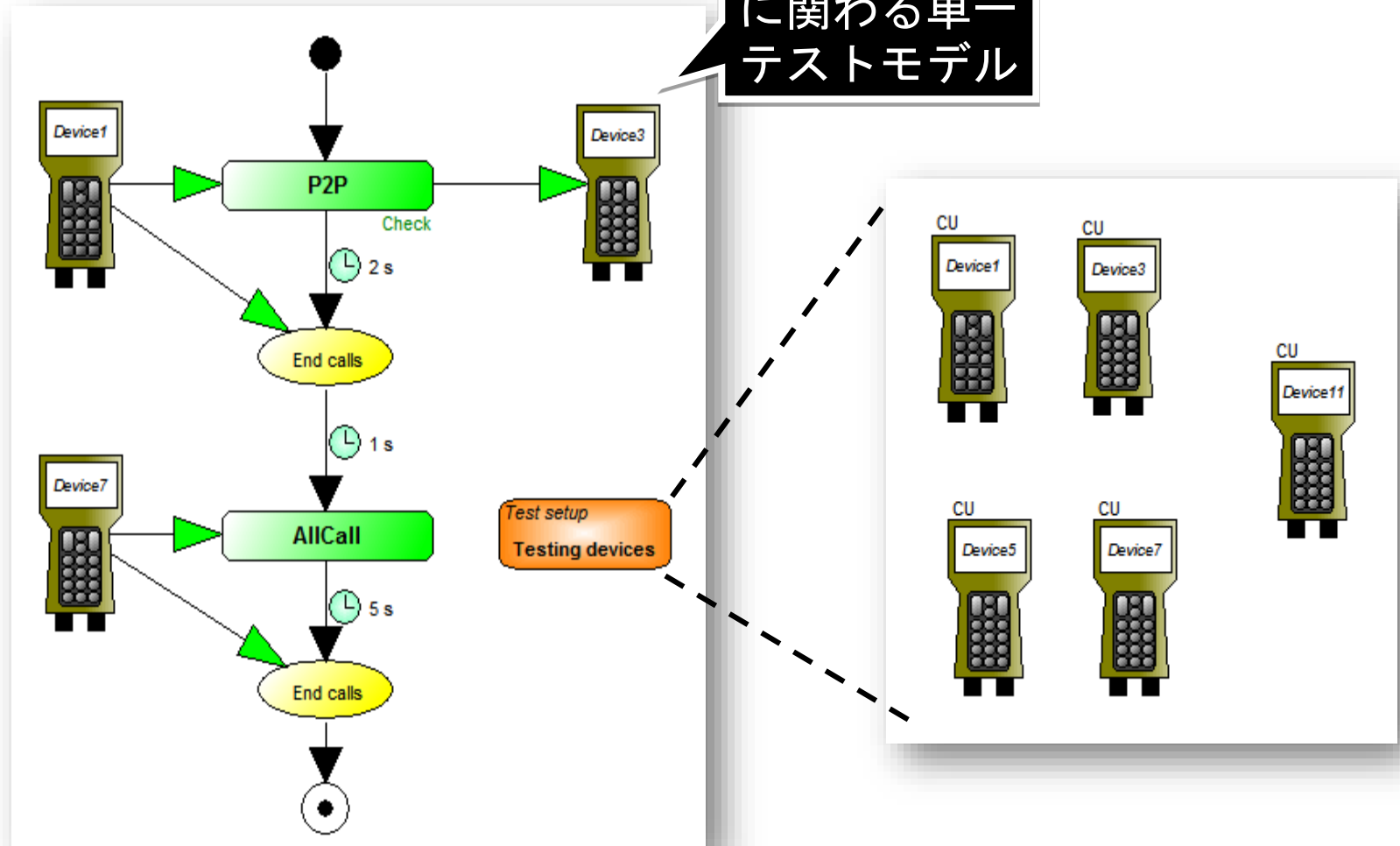
2つの言語 比較





Model example 1: 単一テストケースごとにモデル化

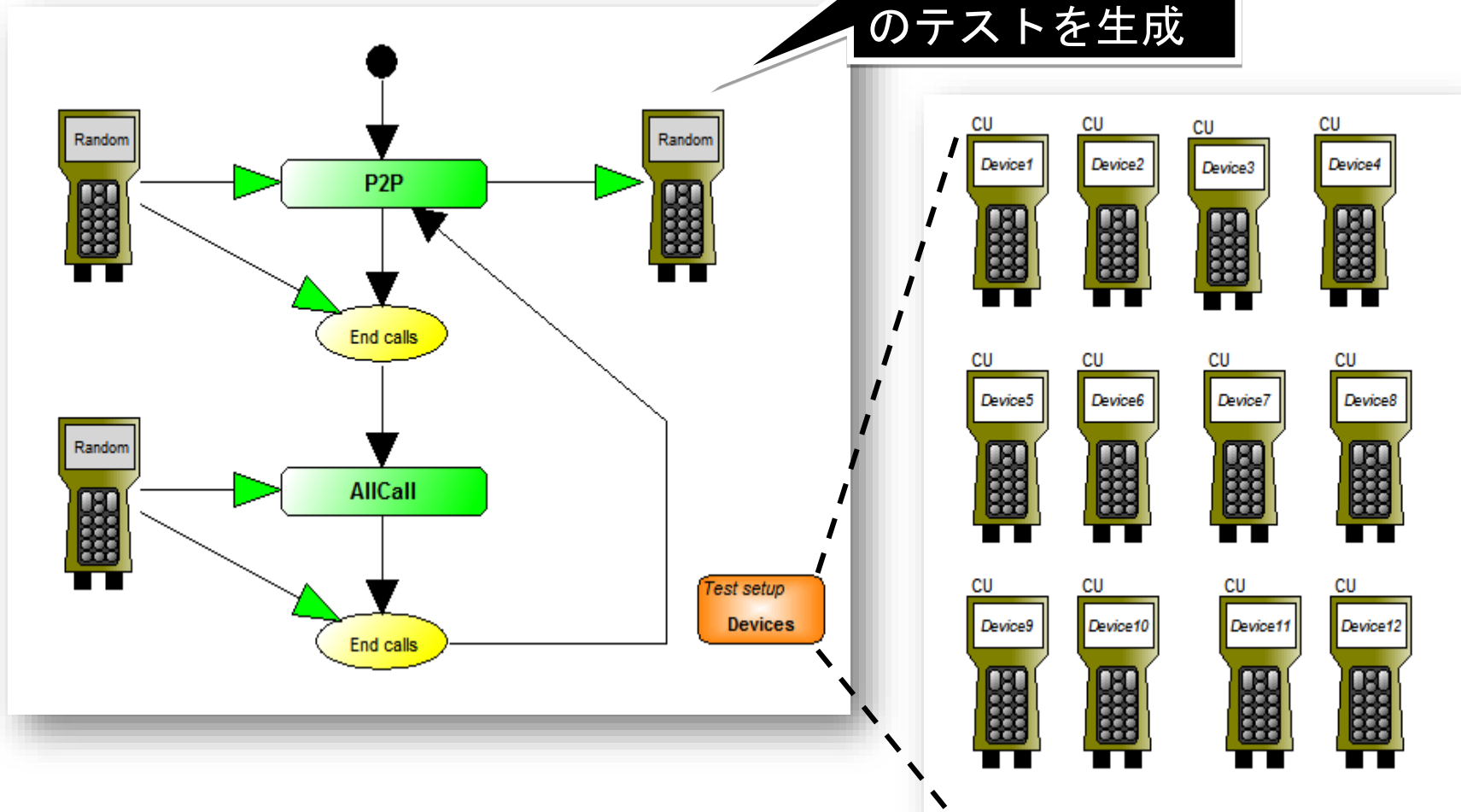
Device1,3,7
に関わる単一
テストモデル





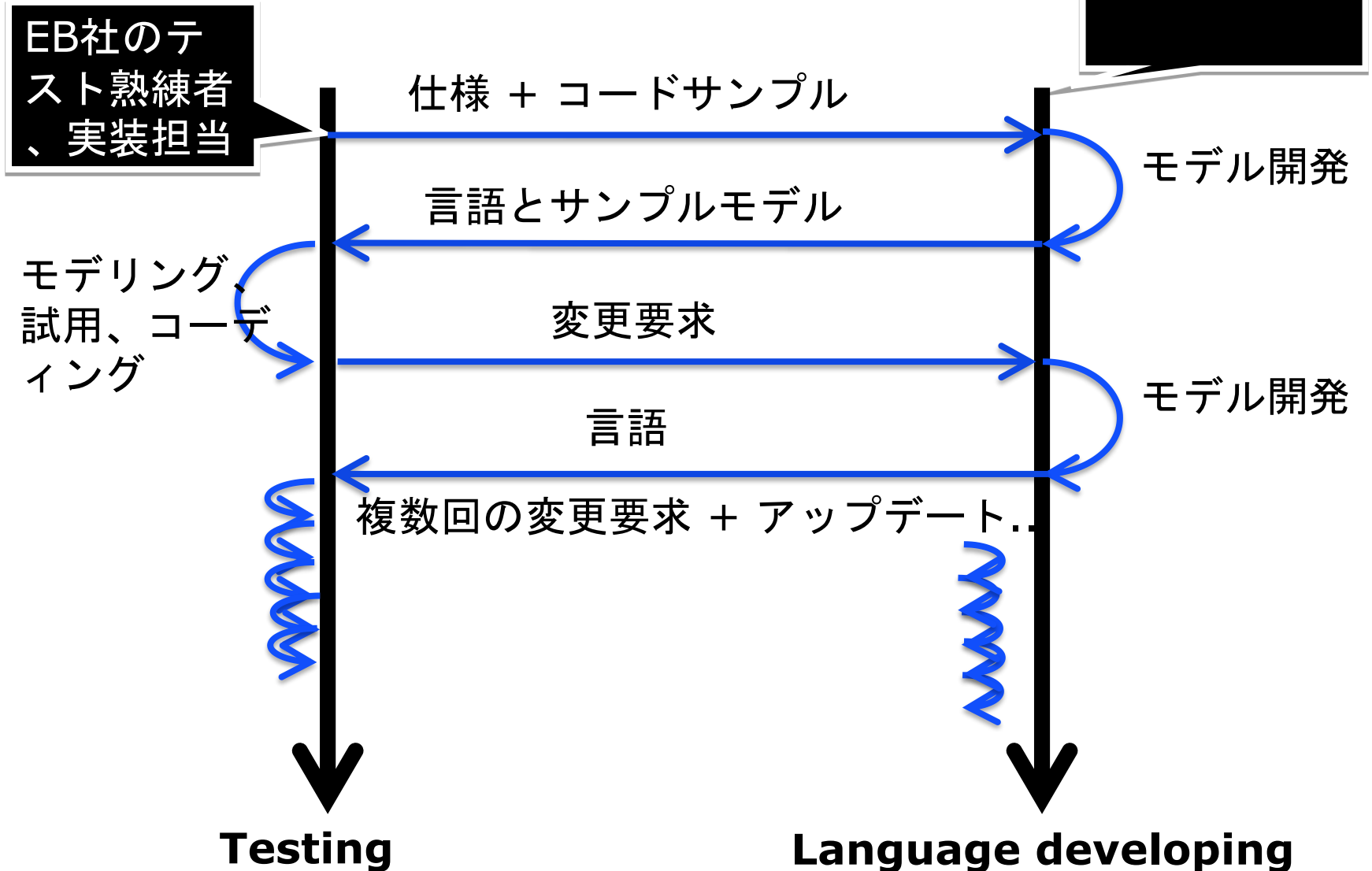
Model example 2: テストのロジックをモデル化

全てのDeviceを
Randomに複数の
テストを生成





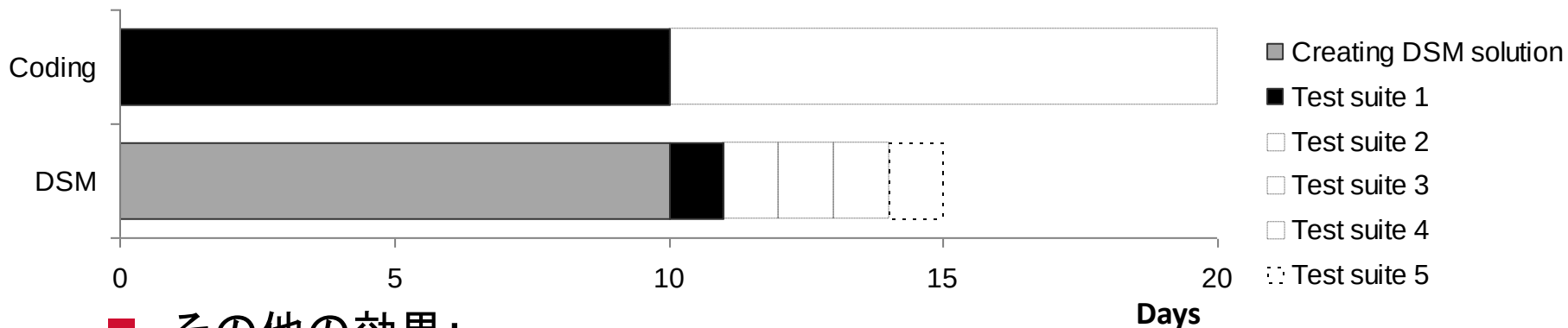
言語開発





成果

- モデリングによって x 10 倍速くなった
- DSM言語開発期間:
 - 最初のバージョンに2 週間
 - 追加1 週間で更なる改善



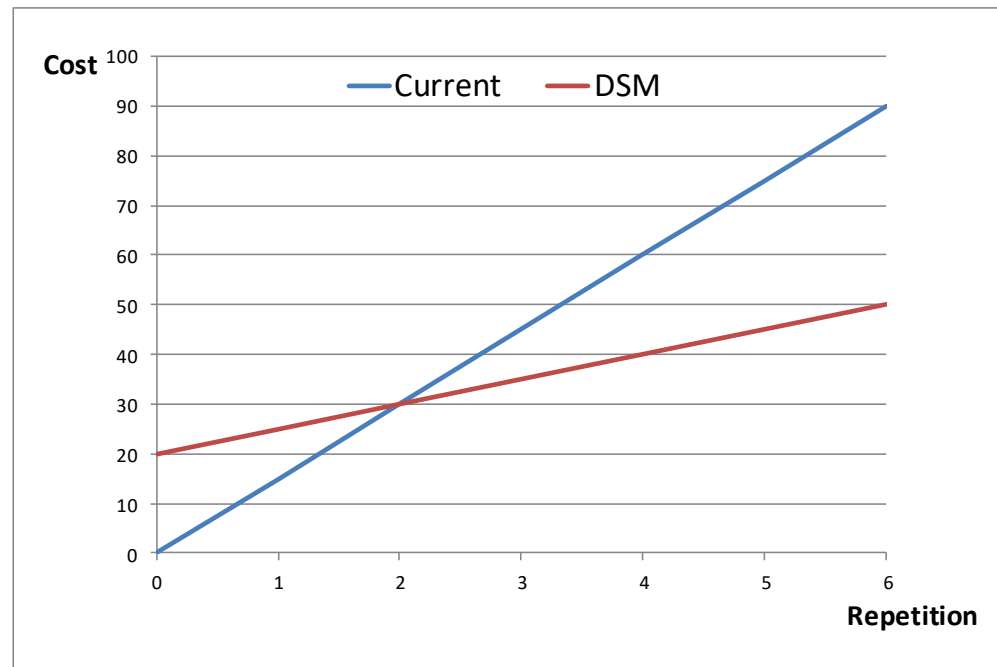
- その他の効果:
 - 視覚化により、わかりやすくなった
 - テストの構成が簡単になった
 - MBTによってテストのカバレッジを飛躍的に改善
 - MBT モデルによって大量のテスト実行
 - テストケース生成に特別なスキルが不要になった



DSM による経済効果

■ Repetition（繰り返し活用することで）：

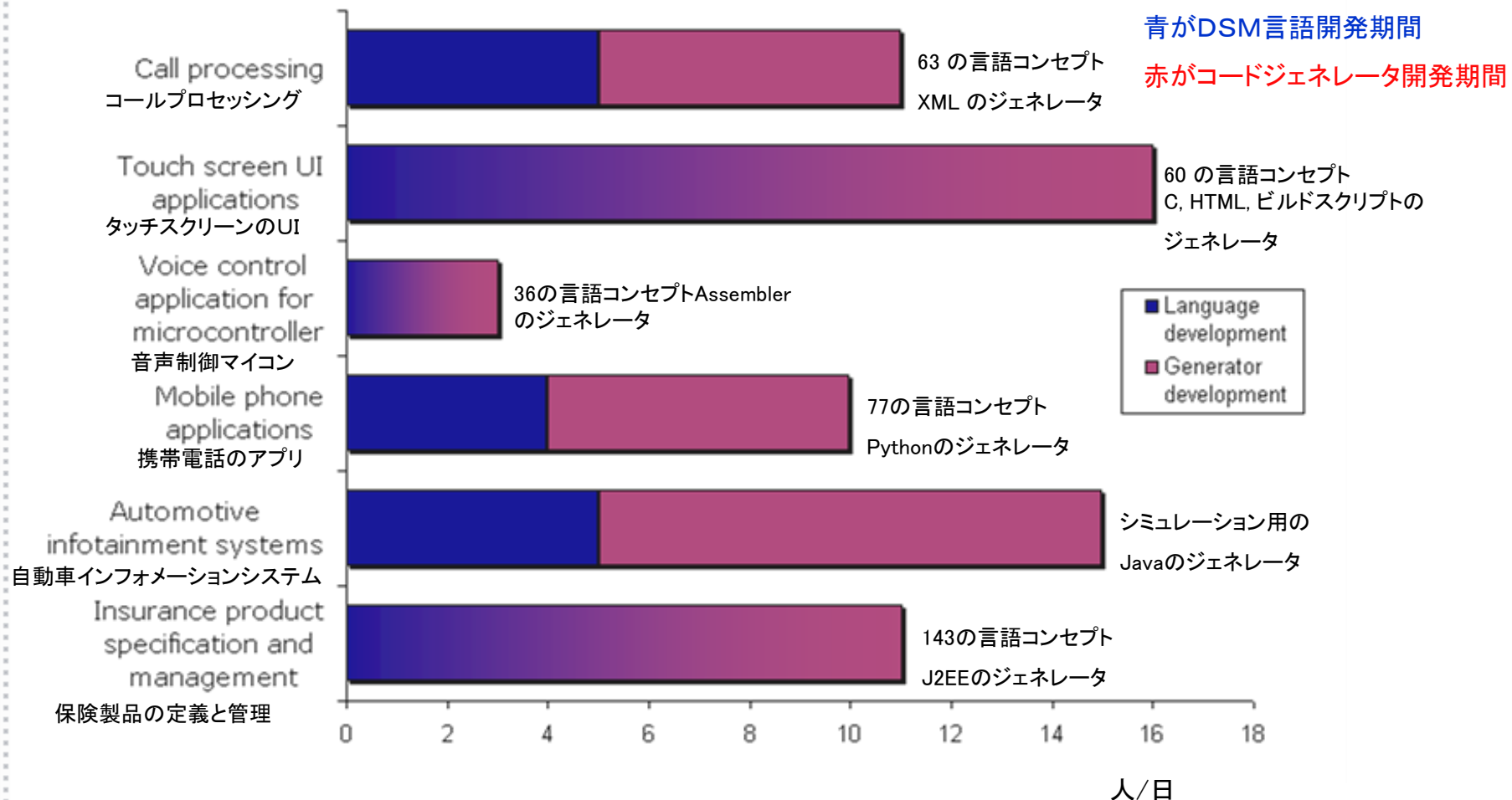
- 製品バリエーションの数
- 同等機能（フィーチャ）の数
- 開発者数
- ドメインの熟練者へのアウトソース





MetaEdit+ ならDSM構築に要する期間は数週間

DSM開発には数年・数億円かかると言われてきたが



<https://www.fuji-setsu.co.jp/products/MetaEdit/blogs.html>



What engineers say



“DSM環境の構築に時間をかける必要はなかった。1人の熟練者によって数週間で用意された後、他の開発者は皆、熟練者のように開発できるようになった” Antti Raunio



“MetaEdit+ を活用して、1週間で DSL 環境を構築できた”, Jukka Manninen



"MetaEdit+ によるモデル化は、Eclipse GMF に比べて非常に簡単で効果的であった" Ulf Hesselbarth



“MetaEdit+ は最もフレキシブルなツールであり、独自の言語シンタックスを直ちに定義できた”, David Narraway

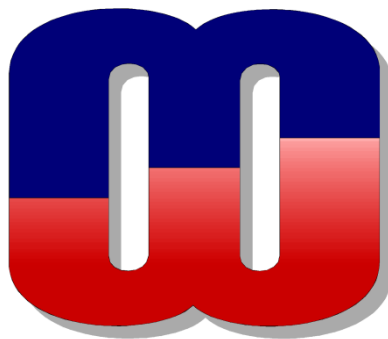


まとめ

- Domain-Specific Modeling 言語:
 - 生産性の向上
 - 品質の向上
 - 新人でも容易に活用できるようになる
- プロセスの変更など開発ライフサイクルの話題について言及しませんでした、:
 - 設計者、開発者が同じ言語（モデル）を扱える
 - テストデータもモデルから生成できる
 - 各種ドキュメントも自動生成
- 言語を構築するのに多くの時間を要しない！



ありがとうございました!



info@metacase.com
www.metacase.com

Europe:
MetaCase
Ylistönmäentie 31
FI-40500 Jyväskylä, Finland
Phone +358 14 4451 400
Fax +358 14 4451 405



FUJI SETSUBI

販売代理店: 富士設備工業株式会社 電子機器事業部
TEL: 072-252-2128 <https://www.fuji-setsu.co.jp>



References

- Kelly, S., Tolvanen, J.-P., Domain-Specific Modeling: Enabling Full Code Generation, Wiley, 2008.
<http://dsmbook.com>
- Kärnä, J., et al. Evaluating the Use of Domain-Specific Modeling in Practice, Proceedings of the 9th Workshop on Domain-Specific Modeling, HSE Print, B-108, 2009.
- Puolitaival, O.-P., Home Automation DSL case, Code Generation Conference, 2011
- Puolitaival, O.-P., Kanstren, T., Rytty, V.-P., Saarela, A., Utilizing Domain-Specific Modeling for Software Testing, Proceedings of VALID, October 2011

