12. 応用事例情報

本章では、形式手法の導入を検討する上で参考となる実システムに対する応用事例について、費用対効果や導入時の課題・工夫点等を中心にまとめる。

想定読者	CIO, 組織管理者、開発技術、発注者等
目的	ユーザが関心を持つ類似事例から、形式手法の実用度、費用対効果、課題・工夫点等の情報を提供することにより、形式手法の導入を支援する
想定知識	ソフトウェア開発プロセスの概略
得られる事	● 形式手法に関する実用度および費用対効果 ● 形式手法導入時の課題とその解決策 ● 手法・ツール選択のための指針 ● 形式手法導入時における教育方法の指針

12.1. 事例の整理項目

各事例は、表 12-1 に示す項目で整理した。

表 12-1:応用事例情報の整理項目

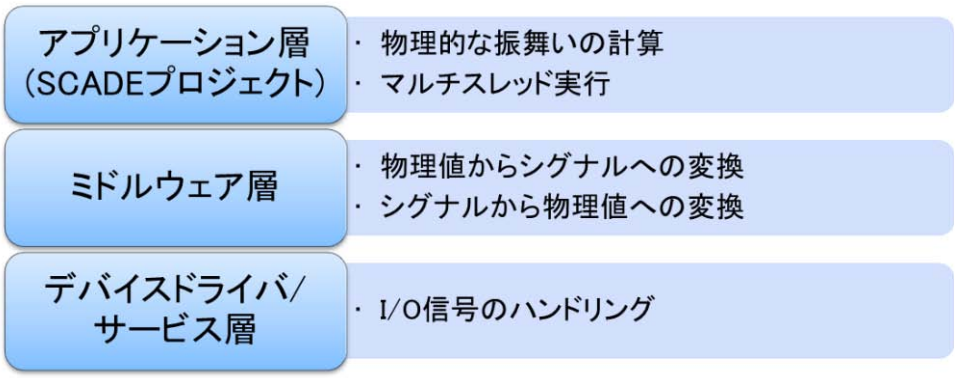
項目	概要
ドメイン	開発対象のドメイン
開発対象	形式手法を利用して開発した対象
国	開発された国
開発組織	開発を行った組織
形式手法(言語、ツール)	利用した形式手法の言語およびツール
適用範囲・規模(形式手法)	形式手法を適用した規模、全体のシステムに対する割合
適用対象のソフト種別	制御系、リアルタイム制御系、エンタプライズ系の区別
適用目的・工程	形式手法を適用した工程
実装言語	実装に利用した言語
規模(実装)	開発対象の実装コード行数
効果	形式手法を利用して得られた効果
検証内容	検証性質の具体例
検証規模	証明課題、要求仕様に対する検証率
期間	開発期間
形式手法を利用した動機	形式手法を利用して開発を行った理由
手法・ツール選択理由	具体的な手法・ツールを選択した理由
障害と工夫	導入および開発時に生じた障害とその対策方法
体制	開発を行った組織体制
教育	形式手法に関して開発メンバーに対して行った教育
その他(外部リソース他注目点)	外部リソースの利用等、その他注目すべき点
参考資料	参考文献

12.2. 事例の調査結果

以下に形式手法を利用した応用事例を示す。

12.2.1. 富士重工業 - ECU ソフトウェア開発

ドメイン	自動車
開発対象	モータ制御 ECU
国	日本
開発組織	富士重工業
形式手法(言語、ツール)	SCADE
適用範囲・規模(形式手法)	制御ソフトウェアをアプリケーション層、ミドルウェア層、デバイスドライバ/サービス層の 3 層に分け、このうちアプリケーション層全体に対して SCADE を適用した。
適用対象のソフト種別	リアルタイム制御系
適用目的・工程	設計、自動コード生成
実装言語	C
実装規模	不明
効果	ヒューマンエラーの排除及び生産性の向上



(出典: Masaru Kurihara, Automotive ECU software development with SCADE Suite, 2009, http://www.esterel-technologies.com/files/Automotive_ECU_Development-2010.pdf より作成)

図 12-1:ソフトウェアアーキテクチャ設計

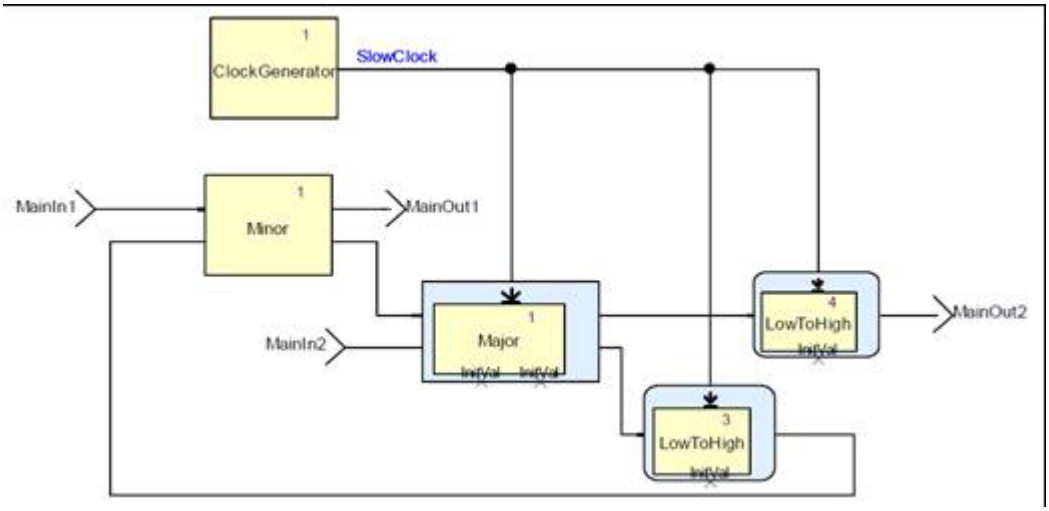
- 詳細情報
  - 検証内容
    - 異なるスレッド間におけるデータハンドリングについて一貫性が保証されることを検証した。
- 判断
  - 形式手法を利用した動機
    - 製品開発時間の短縮に対応する必要があり、従来の開発手法とは異なる開発方法が要求された。
  - 障害と工夫
    - モデルベース開発の導入をスムーズに行うため、既存のソフトウェアアーキテクチャを再設計して 3 層に分け、上層のアプリケーション層に SCADE を適用した。アプリ

ケーション層からは、ターゲットハードウェアや OS を隠蔽した。
- ✧ アプリケーション層と第 2 層のミドルウェア層とのインタフェースを人手で記述するとエラー混入の可能性が生じるため、スクリプトを作成して SCADE 内で実行することにより、インタフェース部分を自動生成した。
- ✧ 開発初期はミドルウェアやハードウェアの仕様も頻繁に変更されるため、ミドルウェア層以下で定義され、アプリケーション層でも利用されるデータ型を手動で管理するとエラーが混入する可能性が高まる。そこで、ミドルウェア層で利用されるデータ型を自動的に抽出し、型定義ファイルを SCADE に提供する Td スクリプトを作成し、実行した。
- ✧ SCADE はマルチスレッドのモデリングを対象としていないため、SCADE 言語を十分に理解してマルチスレッドに適用可能な安全なコードを自動生成した。

・ 情報源
- ➢ CDAJ CAE Solution Conference 2008 | 講演概要, ECU ソフトウェア開発における SCADE の適用例（富士重工業）, http://www.cdaj.co.jp/ccsc2008/lecture/scade_04.html
- ➢ Masaru Kurihara, Automotive ECU software development with SCADE Suite, 2010, http://www.esterel-technologies.com/files/Automotive_ECU_Development-2010.pdf

12.2.2. Intertechnique - 航空機の燃料管理システム

ドメイン	航空運輸
開発対象	Airbus A380 の燃料管理システム
国	フランス
開発組織	Intertechnique, Esterel Technologies
形式手法（言語、ツール）	SCADE
適用範囲・規模（形式手法）	不明
適用対象のソフト種別	リアルタイム制御系
適用目的・工程	設計、自動コード生成
実装言語	不明
実装規模	不明
効果	開発工程の早い段階でシミュレーションによる検証を行うことができたため、機能の不具合が改善されたことにより、ハードウェアとの統合にかかる時間を 60%程度削減することができた。



（出典: A. Jean-Louis Camus, Pierre Vincent, Olivier Graff, Sebastien Poussard, A verifiable architecture for multi-task, multi-rate synchronous software, 2008）

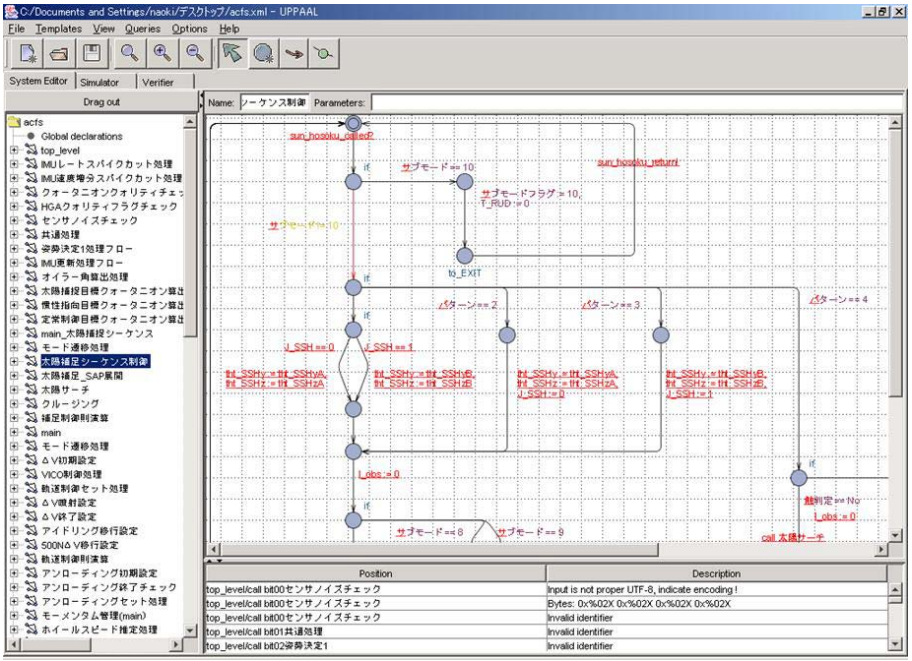
図 12-2：SCADE により作成されたスケジューリングとコミュニケーションモデル

- 詳細情報
  - 検証内容
    - サブシステム間のデータ交換が正しく行われることを検証した。
- 判断
  - 形式手法を利用した動機
    - 安全を保証した設計とその検証を早い段階で行い、組込みソフトウェアのコードを生成する必要があった。
    - DO-178B 標準の要求を満たす必要があった。
  - 手法・ツール選択理由
    - 以前の A340 電子ロード管理システムの実装に成功し、その後 2 年間、Esterel Technologies の協力を得て、SCADE の利用を社内標準とした。
    - SCADE は DO-178B 標準を満たす C コード生成機能を有している。
    - SCADE は並列動作や機能の依存性の簡潔で明確な表現が可能である。
  - 障害と工夫
    - SCADE は単一スレッドのコードを生成するように設計されているため、当初考案したモデル（図参照）に対応する正しいコードを生成することができなかった。そこで、自動コード生成可能な意味的に同一なモデルを作成した。
- 情報源
  - Intertechnique :: Success Stories, Esterel Technologies, http://www.esterel-technologies.com/technology/success-stories/intertechnique
  - A. Jean-Louis Camus, Pierre Vincent, Olivier Graff, Sebastien Poussard, A verifiable architecture for multi-task, multi-rate synchronous software, 2008

12.2.3. JAXA - 人工衛星の姿勢制御ソフトウェア

ドメイン	航空運輸
開発対象	人工衛星である SELENE および WINDS の姿勢制御ソフトウェア
国	日本
開発組織	JAXA (旧 NASDA)
形式手法（言語、ツール）	UPPAAL

適用範囲・規模(形式手法)	不明
適用対象のソフト種別	リアルタイム制御系
適用目的・工程	仕様記述
実装言語	C
実装規模	不明
効果	時間制約を考慮した検証を行うことができた。



（出典: Naoki Ishihama, Hideki Nomoto, Haruka Nakao, NASDA IV&V, NASA IV&V Facility, 2003）

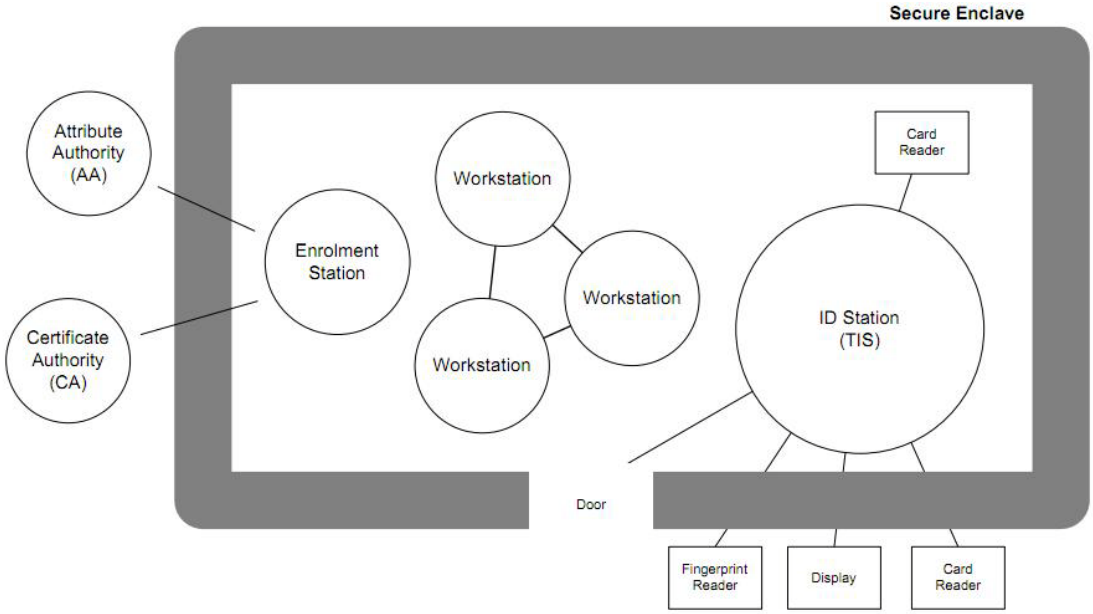
図 12-3：UPPAAL によるシーケンス制御モデル図

- 詳細情報
  - 検証内容
  - モデルの不変条件や到達性を検証した。
- 判断
  - 形式手法を利用した動機
    - 高信頼のシステムソフトウェアを実現する必要があった。
  - 手法・ツール選択理由
    - ソフトウェアの要求がデータフローチャートで記述されており、UPPAAL モデルはデータフローチャートであるから、UPPAAL の適用を決定した。また、時間制約を記述する必要があり、UPPAAL は時間オートマトンを拡張したネットワークをモデル化することができる。
- 組織
  - 体制
    - ソフトウェア成果物の妥当性を検証するチームが IV&V(独立検証および妥当性確認)の一環として実施した。
- 情報源
  - Naoki Ishihama, Hideki Nomoto, Haruka Nakao, NASDA IV&V ....., NASA IV&V

Facility, 2003, http://www.nasa.gov/centers/ivv/ppt/172865main_NASDA.ppt

12.2.4. NSA - バイオメトリクス ID 認証を利用した入退室管理システム

ドメイン	その他
開発対象	バイオメトリクス ID 認証ツールのアクセス管理等に利用されるセキュリティソフトウェアである Tokeneer を利用した入退室管理システム
国	米国
開発組織	Praxis High Integrity Systems
形式手法（言語、ツール）	Z (fuzz type checker)、SPARK (SPARK Examiner)
適用範囲・規模（形式手法）	Z 及び英語テキスト: 形式仕様・・118 ページ セキュリティプロパティ・・11 ページ 形式設計・・171 ページ
適用対象のソフト種別	制御系
適用目的・工程	仕様記述、設計、コード検証
実装言語	SPARK
実装規模	宣言・・4964 実行行数・・4975 SPARK flow アノテーション・・6036 SPARK proof アノテーション・・1999 コメント・・8529 合計・・30278
効果	多くの領域で EAL5 の要求以上を達成した。形式手法を利用したことが結果的に効率的であった。



（出典: David Cooper and Janet Barnes, Tokeneer ID Station EAL5 Demonstrator: Summary Report, 2008）

図 12-4: 入退室管理システムの全体像

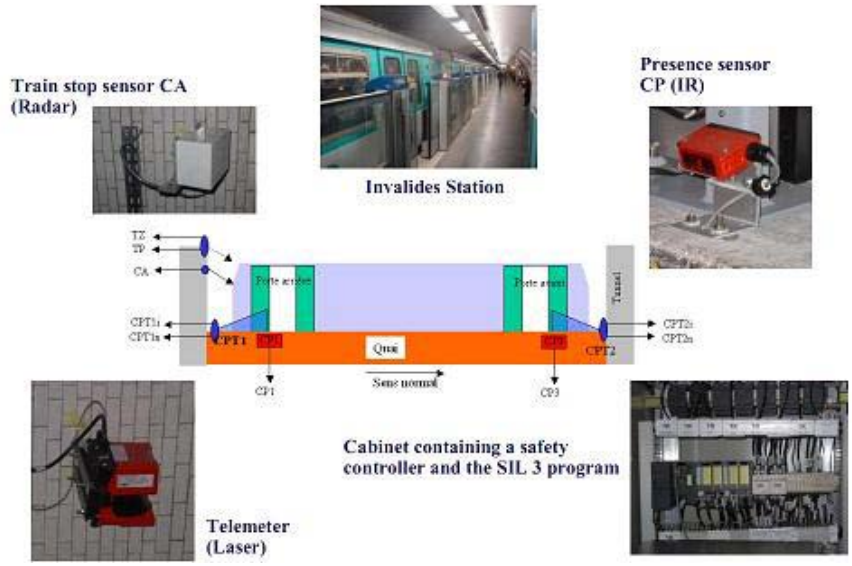
- 詳細情報
  - 検証内容
    - 部屋に入る権限を持つユーザがトークンリーダを通して自分のトークンを TIS (Tokeneer ID Station) に提供したとき、
      もし、ユーザのトークンが正しいものであり、トークン内の認証と認可のデータがユーザの指紋と一致する指紋テンプレートを保有していたら、
      ユーザが部屋に入ったらドアが閉まりロックされ、ユーザのトークンに権限付与証明書が与えられる。
      等。
  - 検証規模
    - SPARK Examiner が生成した Verification Condition292 個のうち、223 個は Examier/Simplifer を利用して自動的に証明された。14 個は Interactive Proof Checker を利用して証明し、55 個はレビューにより証明した。
  - 期間
    - 2003
- 判断
  - 形式手法を利用した動機
    - EAL5 レベルのシステムを、従来の開発プロセスよりもコストを抑えて開発することができるかどうかを実証するために形式手法を利用した。
  - 手法・ツール選択理由
    - Z を利用した理由:
      - notation がチェック可能であり、仕様作成時の小さい不具合も除去できる。
      - 仕様作成時に詳細な設計について考える必要がない。
      - 巨大なシステムを管理可能なサブコンポーネントに分割できる。
      - 型チェックや検証のためのツールサポートがある。
- 組織
  - 体制

- ✧ 3 名の技術者（2 人は Z の読み書きができ、セキュリティ等の専門知識を有していた。1 人は Z の解読はできるが記述スキルは不十分であった）

・ 情報源
  - David Cooper and Janet Barnes, Tokeneer ID Station EAL5 Demonstrator: Summary Report, 2008
  - 形式手法適用調査　調査報告書、情報処理推進機構、2010

12.2.5. ClearSy - パリ地下鉄プラットフォームドアの制御

ドメイン	鉄道
開発対象	プラットフォームドアのコマンドコントローラ
国	フランス
開発組織	RATP、ClearSy
形式手法（言語、ツール）	B (COMPOSYS, B4Free, Atelier B, Brama Animator)
適用範囲・規模（形式手法）	1300 ページのドキュメント セーフティケースは 30 のドキュメントがあり、600 ページ B モデルは 3500 行
適用対象のソフト種別	リアルタイム制御系
適用目的・工程	システム仕様設計、ソフトウェア仕様設計、コード検証
実装言語	LADDAR
実装規模	不明
効果	テスト工程において、ソフトウェアの異常が一つも発見されなかった。



（出典: Guilhem Pouzancre, A Formal Approach in the Implementation of a Safety System for Automatic Control of "Platform Doors", 2006）

図 12-5：プラットフォームドアの構成

・ 詳細情報
  - 検証規模
    - ✧ 1000 の証明責務があり、90%を自動証明、残りを 2 日間で Atelier B interactive

prover を用いて証明した。

- 期間
  - 2006（10 か月）
    - システムアーキテクチャの定義　2 か月
    - システム設計および安全性の実証　4 か月
    - 3 つのプラットフォームへの導入　4 か月

・ 判断
- 形式手法を利用した動機
  - IEC 50126、50128、50129 等に対応する必要があった。
- 手法・ツール選択理由
  - 信頼性の確保および、プロジェクト全体を通してのトレーサビリティの確保のため、プロジェクトのほとんどの工程で利用可能である B を利用した。
- 障害と工夫
  - 列車の到着・出発に関してセンサを利用して検知するが、安全性の高いセンサは高価であった。そのため、比較的安価なセンサを利用し、それを多重化することでシステム全体の安全性を高めた。
  - PSD コントローラのアーキテクチャに依存しない過去の機能解析の結果を再利用した。
  - PLC として SIL3 互換の Siemens7 を利用しており、SIL3 を維持するためには Siemens7 の GUI をとおしての LADDER 言語でのプログラミングが要求されていた。そのため、B を LADDAR へ変換する際に時間制約などを検証するための最適化手法などを導入した。

・ 組織
- 体制
  - プロジェクトマネージャ、開発エンジニア、セーフティエンジニア、検証エンジニア

・ 情報源
- Thierry Lecomte, Thierry Servat, Guilhem Pouzancre, Formal Methods in Safety-Critical Railway Systems, 2007, http://deploy-eprints.ecs.soton.ac.uk/8/1/fm_sc_rs_v2.pdf
- Guilhem Pouzancre, A Formal Approach in the Implementation of a Safety System for Automatic Control of ""Platform Doors"", 2006, http://www.composys.eu/resources/AFIS/Clearsy%20AFIS%202006%20Slides_03_05_2006_en.pdf
- 形式手法適用調査　調査報告書、情報処理推進機構、2010

12.2.6. Airbus - 航空機システム

ドメイン	航空運輸
開発対象	Airbus 社の航空機（A340-500/600、A380）における、航空制御システム、fly-by-wire 制御、表示機器、警告およびメインマシン
国	フランス
開発組織	Airbus
形式手法（言語、ツール）	SCADE、Airbus ACG ツールセット（内製ツール、SCADE-KCG 等）
適用範囲・規模（形式手法）	不明
適用対象のソフト種別	リアルタイム制御系
適用目的・工程	仕様設計、詳細設計、自動コード生成

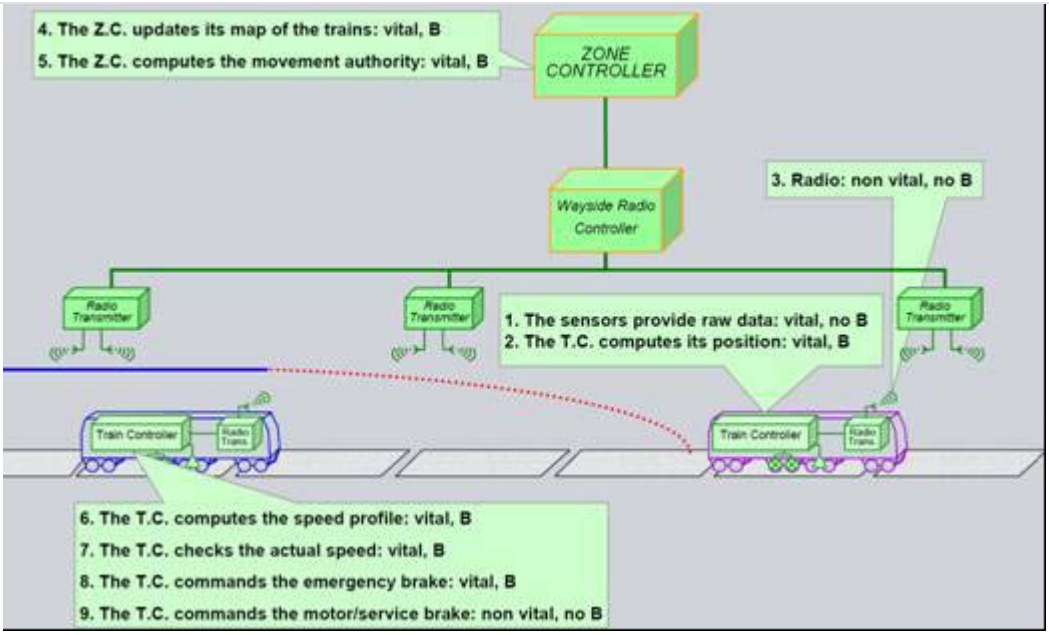
実装言語	C
実装規模	あるコンポーネントにおいて自動生成された C コードは 785,000 行である。
効果	多くのコードが自動生成されたため、コーディングエラーが大幅に削減された。 仕様変更に迅速に対応できた。 トレーサビリティが改善された。 生産性が大幅に改善した。

- 詳細情報
  - 検証規模
    - 各システムの LOC のうち、自動コード生成された割合は、以下のとおりである。
      - 航空制御システム 70%
      - fly-by-wire 制御 70%
      - 表示機器 50%
      - 警告およびメインマシン 40%
  - 期間
    - 1990 年代後半 – 2005
- 判断
  - 手法・ツール選択理由
    - SCADE は、開発者が意識することなく形式手法を利用することができる。したがって、SCADE の操作を習得した開発者がいれば開発可能である。 しかし、生成されたコードの検証を行うためには、数学的素養のあるエンジニアが必要である。
  - 障害と工夫
    - 内製ツールである ACG を利用することで、SCADE KCG で生成された C コードを後処理し、モデルに影響を与えることなくターゲットに最適化されたコードを生成した。
- 組織
  - 体制
    - 重要なコンポーネントは自社開発を行った。サブコンポーネントのうち 1/3 は自社開発であり、残りは社外開発である。
- 情報源
  - 形式手法適用調査 調査報告書、情報処理推進機構、2010

12.2.7. STS - ニューヨーク地下鉄列車制御システムの最新化

ドメイン	鉄道
開発対象	地下鉄カナーシ線列車制御システム
国	米国
開発組織	STS (Siemens Transportation System)、ニューヨーク市都市交通
形式手法（言語、ツール）	B (Atelier B, Edith B)
適用範囲・規模（形式手法）	設計の早い段階で重要な部分とそうでない部分に分け、全ての重要な機能について B を適用した。（ただし、低レイヤの入出力、設定ファイル、main 関数の無限ループを除く。） B の抽象モデルを 125,000 行、具体モデルを 38,000 行手動で作成し、EDiTh B を利用することにより自動的に 110,000 行を生

	成した。
適用対象のソフト種別	リアルタイム制御系
適用目的・工程	仕様設計、詳細設計
実装言語	Ada
実装規模	不明
効果	簡潔で曖昧性の無い、高品質なソフトウェア仕様を作成することができた。 妥当性検証チームは詳細コードの開発ではなく、仕様の開発に集中することができた。



（出典: Daniel Dolle, B in Large-Scale Projects: The Canarsie Line CBTC Experience, Siemens Transportation Systems, http://www.clearsy.com/pdf/b2007_b_in_large_scale_projects_siemens.pdf）

図 12-6：列車制御システムの構成

- 詳細情報
  - 検証規模
    - 証明責務は、抽象モデルが 38,500 個、手動による具体モデルが 19,000 個、自動生成による具体モデルが 25,000 個であった。
  - 期間
    - 2001-2005
- 判断
  - 形式手法を利用した動機
    - パリ地下鉄 14 号線で形式手法（B）を利用した経験があった。
  - 手法・ツール選択理由
    - B を利用することにより、簡潔で曖昧性の無い、高品質なソフトウェア仕様を作成することができる。
    - B はコードが形式仕様のプロパティを維持していることを保証することができる。
    - Atelier B は数万の証明責務を取り扱うことができるほどロバストなツールである。

- 障害と工夫
  - STS で開発された半自動詳細化ツールである EDiTh B を利用することにより、パリ地下鉄 14 号線の開発時と比較して大幅に開発効率が向上した。
- 組織
  - 体制
    - 4 人のチームが 1 年で車載ソフトウェアを開発した。4 人のうち 2 人はパリ地下鉄 14 号線における B を利用した開発の経験者であった。その他 1 人は B の理論についての知識はあるが開発経験はなく、もう一人は B の知識もなかった。形式手法の専門家は必要としなかった。
- 情報源
  - 形式手法適用調査 調査報告書、情報処理推進機構、2010
  - Daniel Dolle, B in Large-Scale Projects: The Canarsie Line CBTC Experience, Siemens Transportation Systems, http://www.clearsy.com/pdf/b2007_b_in_large_scale_projects_siemens.pdf

12.2.8. ClearSy - 北京地下鉄の自動列車停止システム

ドメイン	鉄道
開発対象	地下鉄の自動列車停止システム
国	中国
開発組織	ClearSy、Alstom
形式手法（言語、ツール）	B (Atelier B)
適用範囲・規模（形式手法）	不明
適用対象のソフト種別	リアルタイム制御系
適用目的・工程	仕様設計、詳細設計、自動コード生成
実装言語	Ada
実装規模	不明
効果	仕様が満たされることが数学的に証明された。 リファインメントを繰り返し、最終的に Ada コードを生成した。

- 詳細情報
  - 検証内容
    - 安全が保証されない状態になった際に、緊急ブレーキがかかることを検証した。
  - 期間
    - 2006-2008
- 組織
  - 体制
    - Alstom 社のチームが形式的に仕様設計を行った。この仕様を、ClearSy 社のチームが Alstom 社のチームと連携して B を用いて形式化した。
- 情報源
  - ClearSy 社 Web ページ, http://www.fersil-railway.com/php/projet-urbalis-evolution-en.php

12.2.9. 日立ソリューションズ - Blu-Ray ディスク

ドメイン	その他（家電）

開発対象	Blu-Ray ディスク制御ミドルウェア
国	日本
開発組織	日立ソリューションズ
形式手法(言語、ツール)	SPIN
適用範囲・規模(形式手法)	制御ミドルウェアの 20%程度を抽象化したもの
適用対象のソフト種別	制御系
適用目的・工程	制御アルゴリズム検証・基本設計
実装言語	C
実装規模	200kLOC
効果	• テストでは発見が困難なデッドロックの検出。 • 制御アルゴリズムの性質の保証、想定動作の確認。 • テスト工程半減(2 ヶ月→1 ヶ月)

- 詳細情報
  - 検証内容
    - 要求されたディスク操作は、必ず実行され、待機状態に戻る。
    - デッドロックが存在しない。
    - モジュール間の制御状態のズレが有っても、動作停止などの問題は発生しない。
  - 検証規模
    - 機能要求、安全性に関する4つの検証性質とデッドロックに関して検証した。(100%)
  - 期間
    - 2008 年 10 月～2009 年 3 月
- 判断
  - 形式手法を利用した動機
    - 制御の動作に関して、テストでは保証できない性質に確信を得たかった。
  - 手法・ツール選択理由
    - デバイスを含む並行システムの制御アルゴリズムの検証のため SPIN が適していると判断した。
  - 障害と工夫
    - 制御の基本設計をそのままモデリングすると並行プロセス数が多く状態爆発を発生した。
    - 抽象化のヒューリスティックスを用いて並行プロセスを減らし、対象 API を減らすことにより、API を限定した検証が可能になった。
- 組織
  - 体制
    - システム分析、設計の分析は日立ソフトウェアが実施し、形式記述によるモデリングおよび検証は MRI が実施した。NTT データ、NII、JAIST から、形式モデリングに関する助言を得た。
  - 教育
    - SPIN に関する和書、洋書のテキストを5冊程度精読した。モデリング、検証作業で、オンラインマニュアルなどを利用した。
  - その他
    - NII 等からモデリングにおいて助言を得たことが効果的であった。
- 情報源
  - 本ガイダンス 10 章　ケーススタディ1:ブルーレイディスク

- 石黒 正揮,中島 震,梅村 晃広,田中 一之、組込みソフトウェアの制御機構に対するモデル検査の適用に関する評価実験、コンピュータセキュリティシンポジウム CSS2009
- Masaki Ishiguro, Kazuyuki Tanaka, Shin Nakajima, Akihiro Umemura, Tomoji Kishi, A Guidance and Methodology for Employing Model-Checking in Software Development, APESER: Asia-Pacific Embedded Systems Education and Research Conference, December 14-15, 2009
- 石黒正揮,ソフトウェア開発における形式手法導入に関する課題と解決アプローチ, 先端ソフトウェア工学に関する Grace 国際シンポジウム,形式手法の産業応用ワークショップ 2010,WIAFM2010: Workshop on Industrial Applications of Formal Methods, p.41-48, 2010 年 3 月 15 日, 国立情報学研究所, Grace-TR-2010-03

12.2.10. 日立ソリューションズ - 入退室管理システム

ドメイン	その他(入退室管理)
開発対象	入退室管理システム
国	日本
開発組織	日立ソリューションズ
形式手法(言語、ツール)	SPIN, VDM++
適用範囲・規模(形式手法)	入退室管理システムの制御アルゴリズム
適用対象のソフト種別	制御系
適用目的・工程	制御アルゴリズムの検証、要求の妥当性確認
実装言語	C
実装規模	30kLOC
効果	待合室待ちリストのオーバーフローの可能性の発見。

- 詳細情報
  - 検証内容
    - 待合室に入室した人は、いつか必ず案内される
    - 閲覧室に案内された人だけが、タッチパネル脇 ID 端末で、いつか必ず認証 OK となる
    - 閲覧室から退出した閲覧者は、いつか必ず待機者リストから削除される
    - 閲覧室に案内されていない人が、タッチパネル脇 ID 端末で認証を行っても、常に認証をパスしない
    - 閲覧室 X が空室のとき以外、常に案内は変更されない
    - 待機者リストは常にオーバーフローすることはない
  - 検証規模
    - 要求仕様7件のうち4件検証。その他設計関連の性質2件検証。
  - 期間
    - 2009 年 6 月～2010 年 3 月
- 判断
  - 形式手法を利用した動機
    - 要求仕様の妥当性、運用上の条件などを明確化する。制御アルゴリズムの正しさを検証する。
  - 手法・ツール選択理由
    - 制御系の検証を行うため SPIN を用いた。また妥当性確認のため VDM++を使った。
  - 障害と工夫

- ✧ 時間に関わる性質は扱えなかった。状態爆発が発生したためメッセージの送信のみを行うプロセスを統合するなどして、並行プロセスの数を減らした。

- 組織
  - 体制
    - ✧ システム分析、設計の分析は日立ソフトウェアが実施し、SPIN 形式記述によるモデリングおよび検証は MRI が実施した。VDM++の妥当性確認は NII で実施した。
  - 教育
    - ✧ SPIN に関する和書のテキストを2冊程度精読した。
  - その他
    - ✧ NII 等からモデリングにおいて助言を得たことが効果的であった。
- 情報源
  - 本ガイダンス第 11 章ケーススタディ2: 入退室管理システム
  - 入退出管理システムに関するモデル検証の一例、赤井健一郎 (株式会社三菱総合研究所) 石黒 正揮 (株式会社三菱総合研究所) 中島 震 (国立情報学研究所) 田中一之 (株式会社日立ソリューションズ)

12.2.11. Lockheed Martin、Praxis - C130J ミッションコンピュータ

ドメイン	航空運輸
開発対象	航空制御システム
国	イギリス
開発組織	Lockheed Martin、Praxis
形式手法(言語、ツール)	CoRE、SPARK (SPARK Examiner, SPADE Automatic Simplifier, SPADE Proof Checker)
適用範囲・規模(形式手法)	ソフトウェアのコアとなる部分(全体の 80%)
適用対象のソフト種別	リアルタイム制御系
適用目的・工程	要求仕様(CoRE)、コード記述・検証 (SPARK)
実装言語	SPARK
実装規模	200K 行程度
効果	人手で行う場合は 30 日かかるようなバグの発見が、SPARK Examier を利用することにより 1 時間で発見できた。

- 詳細情報
  - 検証内容
    - ✧ CoRE で記述された仕様とコードから、SPARK で Proof Obligation を生成し、検証を行った。
    - ✧ 多くの証明責務は SPADE Automatic Simplifier を用いて証明できた。残りは SPADE Proof Checker を用いて対話的に証明した。
- 判断
  - 形式手法を利用した動機
    - ✧ DO178B のレベル A への準拠が要求されていた。ここでは、MC/DC テストカバレッジが要求されている。このテストは多大な時間を要するため、出来るだけ 1 回限りの実行で終わらせたいという動機があった。そのため、MC/DC テストを実行するまえに、全てのバグを発見しておく必要があった。
  - 手法・ツール選択理由
    - ✧ 本ドメインにおいて、SPARK は多くの導入事例があったため。

- ✧ ソフトウェアに多くの入力と出力がある場合は、CoRE が適合しているため。
- ➢ 障害と工夫
  - ✧ 効率よく検証を行うため、コードの構造をできるだけ単純化した。たとえば、事前定義のチェックなどをコードから削除した。
  - ✧ プログラムユニットの実装及び検証を別々に実施した。

・ 情報源

- ➢ Roderick Chapman, Industrial Experience with SPARK, ACM SIGAda Ada Letters - special issue on presentations from SIGAda 2000 Martin Croxford and James Sutton, Breaking Through the V and V Bottleneck, Lecture Notes in Computer Science, Vol. 1031, 1996

12.2.12. イギリス国防省 - SHOLIS(ヘリコプタ着陸システム)

ドメイン	航空運輸
開発対象	艦載ヘリコプタ運行システム
国	イギリス
開発組織	PMES (Power Magnetics and Electronic Systems)、Praxis Critical Systems
形式手法(言語、ツール)	Z (CADiZ tool)、SPARK (Examiner, Simplifier, Proof Checker)
適用範囲・規模(形式手法)	要求ドキュメントは 4000 行以上。ソフトウェア要求仕様は Z、英語、数学的定義を含めて 300 ページ。
適用対象のソフト種別	リアルタイム制御系
適用目的・工程	要求仕様(Z)、設計(Z、SPARK)、コード記述(SPARK)
実装言語	SPARK
実装規模	133000 行(Ada 宣言 13000 行、Ada 文 14000 行、SPARK flow アノテーション 54000 行、SPARK proof アノテーション 20000 行、空白行やコメント 32000 行)
効果	UK Interim Defense Standard (IDS) 00-55 及び 00-56 への準拠

・ 詳細情報

- ➢ 検証内容
  - ✧ 最も重要な Safety プロパティは「あるセンサの値が現在の SHOL (Ship Helicopter Operating Limits, 操作を行う際の許される値) の範囲を逸脱していた場合は警告が出され、センサの値が SHOL の範囲に収まっていた場合は警告が出されない」ということであり、これは Z を用いて証明された。
  - ✧ グローバル変数や定数の一貫性、初期状態の存在、事前条件の検査について Z を用いて実施した。
  - ✧ 障害発見比率は以下のとおりである。(障害発見比率/工数比率)を計算すると、Z による証明の工程が、最も効率的に障害を発見していることがわかる。

工程	障害発見比率 [%]	工数比率 [%]
仕様	3.25	5
Z による証明	16	2.5
概念設計	1.5	2
詳細設計、実装、簡易テスト	26.25	17
単体テスト	15.75	25
統合テスト	1.25	1
コードの証明	5.25	4.5
システム妥当性テスト	21.5	9.5
受け入れテスト	1.25	1.5
その他（スタッフの教育、プロジェクト管理と計画等）	8	32

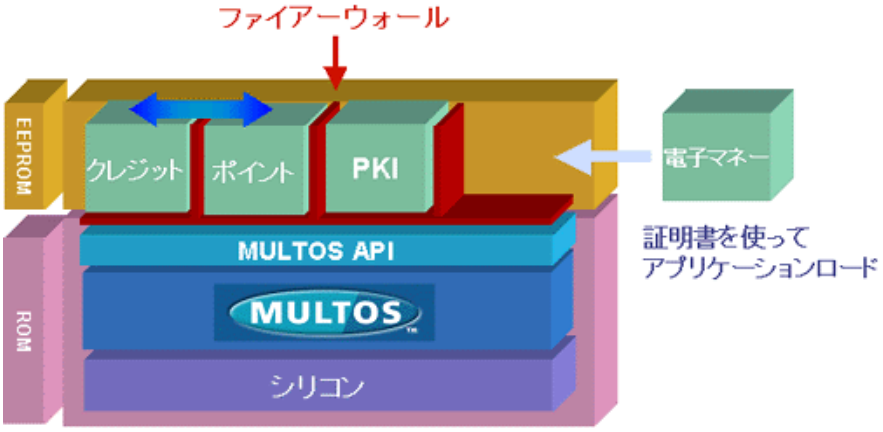
  - 期間
    - 1996 年 - 1997 年

- 判断
  - 形式手法を利用した動機
    - UK Interim Defense Standard (IDS) 00-55 及び 00-56 に準拠する必要があった。これは、形式的な安全管理及び品質システム、システムの振る舞いの形式仕様、仕様及びコードレベルの両方における形式検証、プログラムプロパティ（インフォメーションフロー、タイミングやメモリ使用量等）の静的解析を要求する。
    - テストを実施するよりも、形式手法による検証を実施するほうがコスト面で有利であった。
  - 手法・ツール選択理由
    - SPARK を利用した理由は、論理的な健全性、簡潔な形式記述、表現力、セキュリティ、検証性、実行時の時間と使用リソース量である。
- 組織
  - 体制
    - 検証は 4 人の技術者によって実施された。二人は Z の検証を担当した。このうちの一人と残り二人は、Z の仕様に基づいて SPARK 検証アノテーションを生成し、SPARK の検証を実施した。
    - 全ての技術者は、Z や CSP 等の形式手法に数年間関わっていた。しかし、SPARK Simplifier や Proof Checker を使ったことがあるのは一人だけであった。
- 情報源
  - Steve King, Jonathan Hammond, Rod Chapman, and Andy Pryor, Is Proof More Cost Effective than Testing?, IEEE Transactions on Software Engineering, Vol. 26, No 8, August 2000

12.2.13. Mondex International、Altran Praxis - MULTOS 認証システム(Global Key Center)

ドメイン	通信
開発対象	MULTOS (IC カードのプラットフォーム) の認証局 (IC カードの活性化等を行う)
国	イギリス
開発組織	Mondex International、Altran Praxis
形式手法（言語、ツール）	Z、CSP、SPARK (Examiner)
適用範囲・規模（形式手法）	重要な部分のみ形式手法を適用。28 のセキュリティポリシモデル

	があった
適用対象のソフト種別	エンタプライズ系
適用目的・工程	要求仕様(Z)、設計(Z->CSP)、自動コード生成(CSP->Ada)、コード検証(Ada に対して Spark Examiner の利用)
実装言語	SPARK(30%)、Ada95(30%)、C++(30%)、C(5%)、SQL(5%)
実装規模	10 万行程度
効果	実装前に問題点を理解することができる 各工程をより厳密に進めることができる 可用性 99.999%を達成 形式手法を用いないで開発したシステムよりも、バグの数を少なく抑えることができた。(2002 年時点でバグは 4 つのみ発見されている)



(出典:「MULTOS について:IC カードシステムソリューションズ:日立」,
http://www.hitachi.co.jp/Prod/comp/ic-card/about_ic/multos.html)

図 12-7:MULTOS の概要

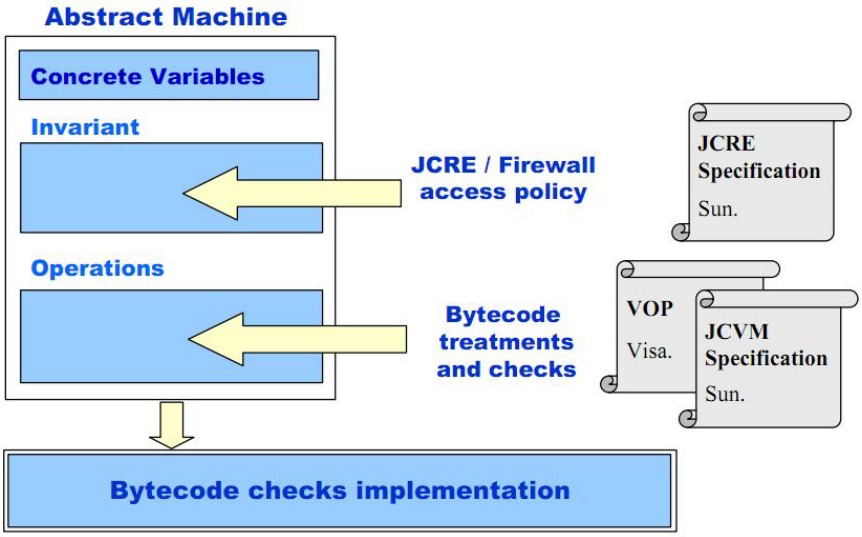
- 詳細情報
  - 検証内容
    - CSP モデルに対して、デッドロックがないこと、セキュリティが特に重要な機能において並行プロセスが存在しないことを検証した。
    - SPARK のインフォメーションフロー解析を実施した。
    - Communications-Electronics Security Group のマニュアルである「F」を簡潔にして実践した。
- 判断
  - 形式手法を利用した動機
    - UK ITSEC スキームの最も高いレベル(E6)に準拠する必要があった。E6 では、初期段階から形式手法を使うことを求めている。
  - 手法・ツール選択理由
    - SPARK は、ITSEC E6 の要求(プログラミング言語は、ソースコード中の全ての文に対し不明瞭でない意味を定義することが求められる)を満たし、実際に産業界で利用可能なほぼ唯一の言語である。
    - 実装すべきセキュリティモデルの多くが、Z を利用することで直接モデル化可能であった

- 障害と工夫
  - ✧ 既存製品を利用するとセキュリティ検証が煩雑になるため、できるだけ自前で実装した。
  - ✧ システムを、セキュリティが特に重要な部分、セキュリティに関連する部分、セキュリティとは無関係な部分に分割した。セキュリティが特に重要な部分はSPARKを用いて開発を行った。
  - ✧ CSPのモデルをコードに変換するルールを作成し、適用した。
- 情報源
  - Anthony Hall and Roderick Chapman, Correctness by Construction: Developing a Commercial Secure System, IEEE Software, Jan/Feb 2002, pp18-25

12.2.14. Gemplus - スマートカード

ドメイン	電子部品・デバイス
開発対象	スマートカードのOS
国	フランス
開発組織	Gemplus
形式手法（言語、ツール）	B (Atelier B)
適用範囲・規模（形式手法）	一部のコンポーネントにのみ適用
適用対象のソフト種別	制御系
適用目的・工程	設計、コード生成
実装言語	不明
実装規模	不明
効果	最終的な開発工数を削減することができた。

	形式手法を利用した開発	従来型の開発
開発	12 週間	12週間
検証	6 週間	NA
テスト	1 週間	3 週間
統合	1 週間	2 週間
合計	20 週間	17 週間
レビューによるバグの発見数	13	24
証明によるバグの発見数	29	NA
テストによるバグの発見数	32	71
合計	74	95



（出典: Industrial Use of Formal Methods at Gemplus, Gemplus Research Lab, http://www.gemplus.com/smart/rd/publications/pdf/Lan99fmg.pdf）

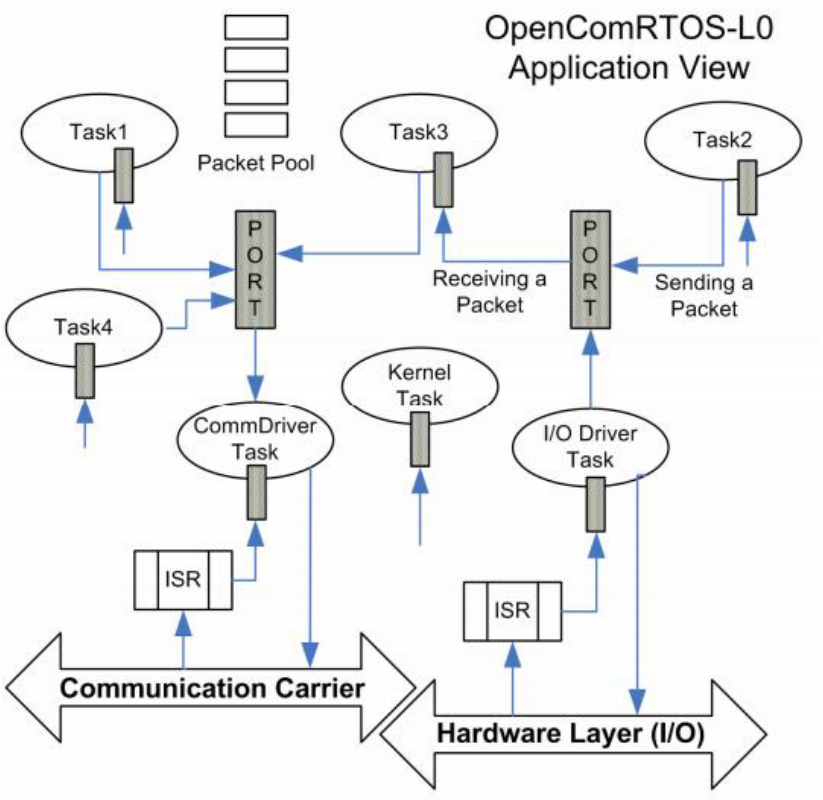
図 12-8:実践した形式モデリングの流れ

- 詳細情報
  - 検証内容
    - ある Java アプレットが他の Java アプレットのデータにアクセスできないこと
    - Java アプレットが OS のコードにアクセスできないこと
    - 正しいモデルを作成し、Atelier B で自動証明を行った
    - 残った証明に関しては、対話式証明を行った。対話式証明により、モデルやループにおける不変条件の不備やそれらをどう構築すれば良いかを理解することができた
- 判断
  - 形式手法を利用した動機
    - スマートカードの高いレベルでの認証が必要だった。また、検証コストを削減する必要があった。
  - 障害と工夫
    - 最初の B モデル仕様のレビューをしっかり行った
    - Atelier B の自動証明に適合するようにモデルをチューニングした
- 情報源
  - Jean-Louis LANET, The use of B for Smart Card, In Forum on Design Languages, 2002

12.2.15. Altreonic - 分散 RTOS (OpenComRTOS)

ドメイン	電子部品・デバイス
開発対象	組込み機器用リアルタイム OS
国	ベルギー
開発組織	Altreonic
形式手法（言語、ツール）	TLA+/TLC
適用範囲・規模（形式手法）	一部分のみに適用
適用対象のソフト種別	リアルタイム制御系
適用目的・工程	アーキテクチャ設計

実装言語	不明
実装規模	不明
効果	RTOS の性質である安全性とリアルタイム性を実現できた コードサイズが従来と比較して1/10 程度になった



（出典: Eric Verhulst, Gjalt de Jong, OpenComRTOS - Distributed RTOS development using formal modeling methods）

図 12-9：OpenComRTOS のアプリケーション概要

- 詳細情報
  - 検証内容
    - 形式モデルに対し、形式モデルエンジニアとソフトウェアエンジニアがミーティングを行い、より詳細なモデルを作成していった。また、モデルが正しいかどうかのチェックを行った。
    - モデルが十分実装に近くなったら、PC 上の仮想的なターゲット上でシミュレーションモデルを構築した。
    - 次にこのコードを実際の 16 ビットのマイクロプロセッサに移行し、最適化した。
- 判断
  - 形式手法を利用した動機
    - RTOS は高い安全性と信頼性を要求されるにも関わらず、2004 年時点では、ほとんどの RTOS が検証されていなかった。信頼性の高い RTOS を構築するために形式手法の適用を決めた。
  - 手法・ツール選択理由
    - C 言語に近い SPIN 等よりも、より抽象的なタームの重要性を理解することができる。

- 障害と工夫
  - 完全な証明のためにはターゲットとなるハードウェアもモデル化して検証する必要があるが、それは複雑であり状態爆発を起こすため、実施しなかった。
  - TLA モデルはパラメータ化ができないため、特定の部分だけをモデル化した。
- 組織
  - 体制
    - 初期のアーキテクチャを定義した後、2 チーム作成し、1チームはアーキテクチャモデルを作成し、もう1チームは、TLA+/TLC を利用して初期の形式モデルを作成した。
- 情報源
  - Bernhard H.C. Sputh, et.al, OpenComRTOS: Reliable performance with a small code size
  - Eric Verhulst, Gjalt de Jong, OpenComRTOS - Distributed RTOS development using formal modeling methods

12.2.16. National Air Traffic Services (NATS) Praxis - iFACTS、航空管制システム

ドメイン	航空運輸
開発対象	航空管制システムの拡張機能。iFACTS (Interim Future Area Control Tools Support)は、NATS に従来からある航空管制システム NERC (New En-Route Centre)の拡張であり、管制官を支援するツールを提供する。
国	イギリス
開発組織	Praxis
形式手法（言語、ツール）	Z (Microsoft Word Add-ins, Fuzz: 型チェッカー)
適用範囲・規模（形式手法）	機能仕様記述 ATS システムソフトウェアの開発 既存の NERC システムに付加し、仕様記述と実装 4250 ページ
適用対象のソフト種別	エンタプライズ系
適用目的・工程	仕様記述、コーディング、テスト、検証
実装言語	SPARK Ada
実装規模	150,000SLOC (Ada SPARK)
効果	[バグ排除] ソフトウェアに重大な欠陥は発見されず、ソフトウェア品質に問題がないことが確認された。 [形式手法の経験] Z を用いたモデリングを経験： テスト実施者やレビュー実施者はテストケース作成やコード・設計のレビューを効率的に行える　ようになった。 各工程間のコミュニケーション促進・曖昧さを解消に役立った。 （設計者、プログラマ、テスト実施者・コードレビュー実施者、保守作業における人的要素を過小評価しないこと） [規格、認証準拠] IEC61508、SIL4 対応に対応できた。

- 判断

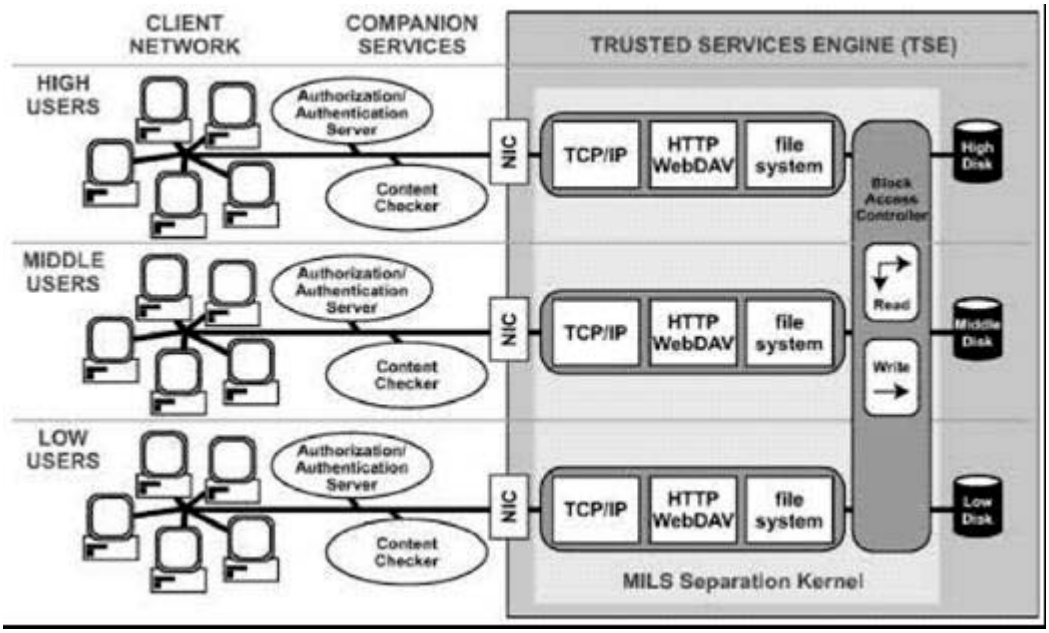
- 形式手法を利用した動機
  - 英国政府は、空港の年間フライト数の急増に備えて、信頼性の高い航空管制システムを導入する必要があった。
- 障害と工夫
  - Z の記述には Word を用いた。言語とツールと同時に新しいことを教えるのは困難という判断があった。Z のフォントと FuZZ タイプチェカーの起動ボタン、仕様と構造の対応を確認する GUI ツールへのリンクなどを備える。

・ 組織
- 教育
  - Z の読み
    - 3 日間のコースを実施。1 週間の業務ですらすらと読めるようになった。 75 名をトレーニングした。
  - Z の書き
    - 3 日間のコースを実施。3 ヶ月の業務ですらすらと書けるようになった。 11 名をトレーニングした。
  - SPARK
    - 57 名をトレーニングした。

・ 情報源
- Open-DO, The Use of Formal Methods on the iFACTS ATC Project (Neil White), http://www.open-do.org/2010/04/20/the-use-of-formal-methods-on-the-ifacts-air-traffic-control-project/

12.2.17. Galois - Trusted Services Engine (TSE)

ドメイン	通信
開発対象	Bell-LaPadula モデルに対応した分散ファイルシステム
国	米国
開発組織	Galois (SPAWAR: Space and Naval Warfare Systems Command, NAS: National Security Agency のファンドを受ける)
形式手法（言語、ツール）	Isabelle
適用範囲・規模（形式手法）	一部のソフトウェアコンポーネントである BAC (Block Access Controller)
適用対象のソフト種別	エンタプライズ系
適用目的・工程	設計
実装言語	C
実装規模	780 行 (C コード)
効果	EAL 6 の認証基準への準拠



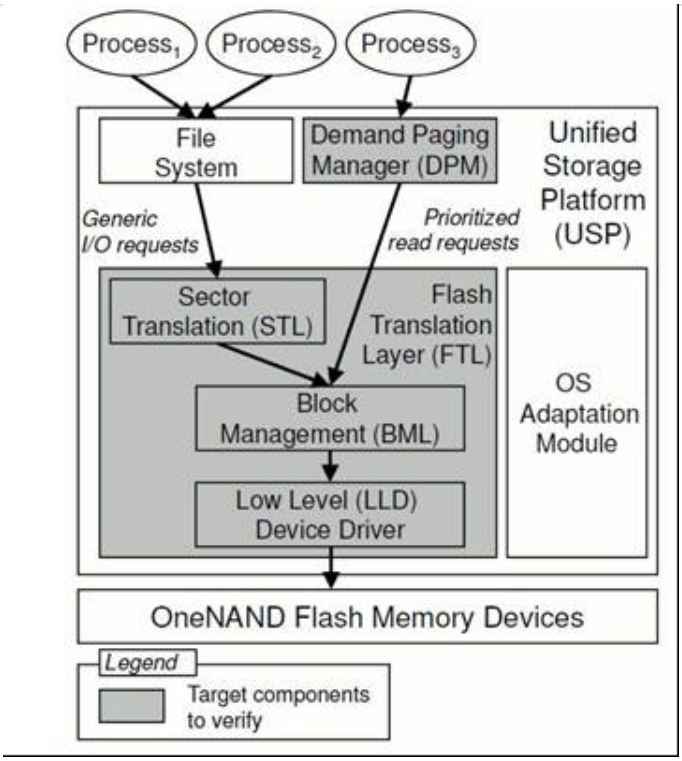
（出典: Christopher Gunderson, Worldwide Consortium for the Grid (W2COG) Research Initiative Phase 1 Final Report, 2006）

図 12-10:TSE アーキテクチャの概要

- 詳細情報
  - 検証内容
    - Isabelle による検証
      - TSE のセキュリティポリシとモデルが正しく対応していること
      - エラー状態へ到達しないこと
      - 高いセキュリティレベルのコンポーネントによるアクションが、低いセキュリティレベルのコンポーネントから不可視であること
    - QuickCheck tool による検証
      - HOL コードと C の実装が正しく対応していること
      - QuickCheck tool により、HOL モデルからテストケースを生成し、C で記述されたコードに対してテストを実施した。
      - 同一レベルで書かれたデータを読めること
      - 正しい読み取りができること
      - 高いセキュリティレベルの読み取りができないこと
      - 高いセキュリティレベルのコンポーネントによるアクションが、低いセキュリティレベルのコンポーネントから不可視であること
    - 人による検証
      - 各ラインについて最低 2 人が HOL コードと C コードを見比べてレビューを行った。
- 判断
  - 形式手法を利用した動機
    - EAL 6 の認証基準に基づいて開発を行う必要があった。
- 情報源
  - Christopher Gunderson, Worldwide Consortium for the Grid (W2COG) Research Initiative Phase 1 Final Report, 2006

12.2.18. Samsung Electronics - フラッシュメモリデバイスドライバ

ドメイン	電子部品・デバイス
開発対象	OneNAND フラッシュメモリ
国	韓国
開発組織	Samsung、KAIST
形式手法（言語、ツール）	BLAST, CBMC
適用範囲・規模（形式手法）	150～2450 (LOC)
適用対象のソフト種別	制御系
適用目的・工程	組み込みシステムの実機テスト前
実装言語	C
実装規模	不明
効果	Samsung が過去に見つけられなかったバグを検出した。 セマフォ例外の不十分なハンドリング 消去時の状態を保持しない論理的なバグ



（出典:Moonzoo Kim, Yunho Kim, Hotae Kim, "A Comparative Study of Software Model Checkers as Unit Testing Tools: An Industrial Case Study," IEEE Transactions on Software Engineering, vol. 99, 2010 ）

図 12-11：OpenNAND フラッシュメモリのためのストレージプラットフォーム

- 詳細情報
  - 検証内容
    - OneNAND フラッシュメモリのプラットフォームソフトウェアの MSR（マルチセクターリード）の動作。

- ✧ レースコンディションが発生しないこと
- ✧ デッドロックが発生しないこと
- ✧ バイナリセマフォにおいて、セマフォの数がどのステップにおいても 0 以上 1 以下であること
- ✧ セマフォを利用している関数の動作が終了したとき、セマフォの数が 1 になっていること

➢ 期間
- ✧ 6 か月

・ 判断
➢ 形式手法を利用した動機
- ✧ バグ検出

➢ 手法・ツール選択理由
- ✧ Blast と CBMC はソフトウェアモデル検査ツールの中では安定している。(10 年近くメンテナンスされている)
- ✧ ユーザコミュニティが存在する。
- ✧ BlastとCBMCはオープンソースソフトウェアである。検証のパフォーマンス向上等、ユーザに改善の余地が残されている。
- ✧ 数年にわたる利用経験

➢ 障害と工夫
- ✧ フラッシュストレージプラットフォームは、物理的なユニットと論理的なユニットを結びつける複雑なデータ構造を多用するため、動作検証が難しい。

・ 情報源

➢ Moonzoo Kim, Yunho Kim, Hotae Kim, ""A Comparative Study of Software Model Checkers as Unit Testing Tools: An Industrial Case Study,"" IEEE Transactions on Software Engineering, vol. 99, no. PrePrints, , 2010

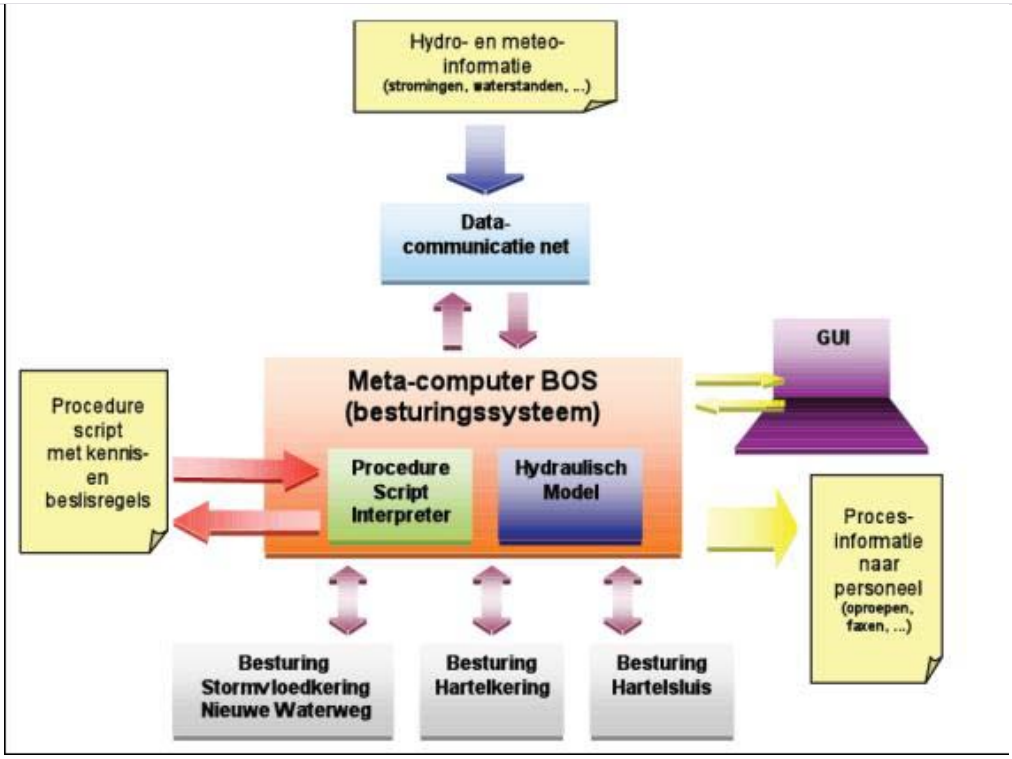
➢ Moonzoo Kim, Yunho Kim, and Hotae Kim. 2008. Unit Testing of Flash Memory Device Driver through a SAT-Based Model Checker. In Proceedings of the 2008 23rd IEEE/ACM International Conference on Automated Software Engineering (ASE '08). IEEE Computer Society, Washington, DC, USA, 198-207. DOI=10.1109/ASE.2008.30 http://dx.doi.org/10.1109/ASE.2008.30

➢ Moonzoo Kim, Yunja Choi, Yunho Kim, and Hotae Kim. 2008. Formal Verification of a Flash Memory Device Driver --- An Experience Report. In Proceedings of the 15th international workshop on Model Checking Software (SPIN '08), Klaus Havelund, Rupak Majumdar, and Jens Palsberg (Eds.). Springer-Verlag, Berlin, Heidelberg, 144-159. DOI=10.1007/978-3-540-85114-1_12 http://dx.doi.org/10.1007/978-3-540-85114-1_12

12.2.19. Logica Nederland B.V. - オランダ運河のマエスラント防潮可動橋（水門）

ドメイン	その他(水門)
開発対象	水門開閉の意思決定システム
国	オランダ
開発組織	Logica Nederland B.V. (旧 CMC Den Haag B.V.) - Rijkswaterstaat (Dutch Ministry of Transport, Public Works and Water management の一部門) の委託を受けて開発
形式手法(言語、ツール)	Z (ZTC type checking tool)、SPIN、PVS
適用範囲・規模(形式手法)	コアとなる部分のみ。Z の仕様記述 - 29 プログラム/20KLOC
適用対象のソフト種別	制御系

適用目的・工程	アーキテクチャ設計（SPIN）、詳細設計（Z）
実装言語	C++
実装規模	C の運用システム - 250KLOC
効果	費用対効果の面で有利であった。 モデル検査により、プロトコルレベルで大きな変更を実施する必要性が判明した Z の仕様記述により、テストケース作成やコード/設計レビューを有効に行うことができた。また、設計者、プログラマ、テスタ、コードレビュー者間の意思疎通を曖昧さを排除して行うことができた



（出典: Klaas Wijbrans, et al., Software Engineering with Formal Methods: The storm surge barrier revisited, ACISION, 2008）

図 12-12：意思決定システム（BOS）の全体像

- 詳細情報
  - 検証内容
    - Promela/SPIN による検証
      - コアアーキテクチャにおいてライブロック及びデッドロックが存在しないこと
    - Z による検証
      - 仕様の妥当性を検証した
      - Ward & Mellor 手法に基づいてモデル化を行った
      - 各プロセス、store、flow における機能やデータをモデル化した
  - 期間
    - 1期: 1995-1996 年、2 期: 2007-2009 年
- 判断
  - 形式手法を利用した動機
    - IEC 61508 では safety-critical システムの開発には形式手法の利用を推奨しており、

本プロジェクトでは IEC 61508 の遵守を決定した

- 組織
  - 体制
    - 1期の開発
      - 3 名の形式手法の専門家の支援を受け、5 名が中心となって開発を行った
      - 7 名が主にコーディングを行った
    - 2 期の開発
      - 7 名が開発を行った
  - 教育
    - Promela/SPIN によるモデリングは習得が容易であった
    - Z によるモデリングは難しく、習得に時間がかかった
- 情報源
  - Ken Madlener, et al., A Formal Verification Study on the Rotterdam Storm Surge Barrier, ICFEM 2010
  - Klaas Wijbrans, et al., Software Engineering with Formal Methods: The storm surge barrier revisited, ACISION, 2008
  - 形式手法適用調査　調査報告書、情報処理推進機構、2010

12.2.20. Rodin Project - 航空情報表示システム

ドメイン	航空運輸
開発対象	CCF 表示及び情報システム (CDIS)
国	イギリス
開発組織	Rodin Project
形式手法（言語、ツール）	Event-B (Rodin)
適用範囲・規模（形式手法）	既存の CDIS システム (1992 年作成) の一部の再構築。 既存の CDIS システムは、VVSL (VDM の一種) で記述され、1200 ページの仕様書、3000 ページの設計ドキュメントが作成された。 再構築された部分における、Event-B の抽象仕様は 4 ページ。
適用対象のソフト種別	エンタプライズ系
適用目的・工程	要求仕様、設計
実装言語	不明
実装規模	不明
効果	[仕様の理解] 抽象的な仕様記述から段階的に詳細化することにより、巨大なシステムの全容の理解を容易にした [バグ排除] 仕様と実際の設計との形式的なリンクを対応させることができた

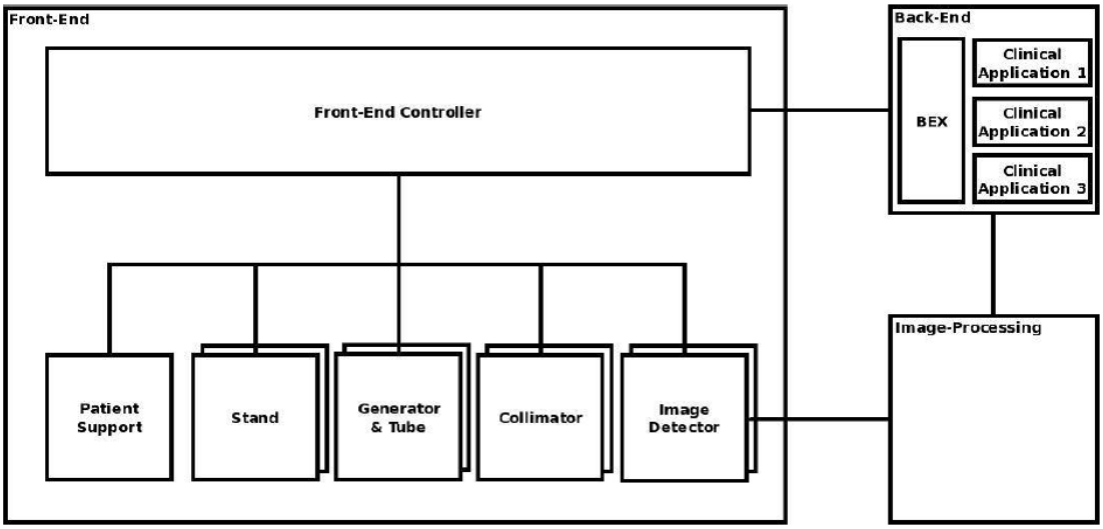
- 判断
  - 形式手法を利用した動機
    - 既存の CDIS システムは VVSL を用いて作成されたが、形式的推論は行われていなかった
  - 手法・ツール選択理由
    - 分散システムモデリングに適しているという理由で Event-B を採用した

- 障害と工夫
  - 理想的な仕様と現実的な設計との間の形式的なリンクの欠如
    - 理想的な仕様の拡張としてまず抽象仕様を作成した。理想的な仕様では、異なるユーザが異なる場所である変化するデータを見たとき、同じ時刻で見た場合は全く同じデータが見えるはずであるが、実際には遅延等によって異なるデータが見えることがある。これを現実的な抽象仕様として記述するために、履歴トラッキングシステムを導入した。システムは全てのデータ変更の履歴を保持し、ユーザは履歴のいずれかのデータを閲覧することとした。この抽象仕様を徐々に詳細化した。
    - まず基本的なディスプレイシステムの仕様記述を行い、航空に特化した部分を全て無視することによって、全体をうまく概観できるようにした。Rodin ツールを利用してそれを徐々に詳細化していった。各ステップの証明責務はおおよそ 20 以下で済んだ。
  - Event-B においてレコード型を SETS、CONSTANTS、PROPERTIES を利用して表現した

・ 情報源
- Rezazadeh, A. and Evans, N. and Butler, M., Redevelopment of an Industrial Case Study Using Event-B and Rodin, BCS-FACS Christmas 2007 Meeting-Formal Methods In Industry.(December 2007)

12.2.21. Philips Healthcare 社 - X 線 CT スキャン

ドメイン	医療
開発対象	X 線 CT スキャン
国	オランダ
開発組織	Philips Healthcare
形式手法（言語、ツール）	Verum ASD Suite
適用範囲・規模（形式手法）	患者をサポートするハードウェアを制御するソフトウェア
適用対象のソフト種別	制御系
適用目的・工程	設計、コード生成
実装言語	C++、C#
実装規模	• C++ 7697 行、C# 19684 行 • 79 人月
効果	[バグ排除] 検知したモデルエラー 423 個 [コスト削減] ASD Suite を利用した開発に要したコストは 541,161 ユーロであり、従来の手法であれば 848,811 ユーロ必要であったと考えられる（実行コード行数より算出）。したがって、36%のコスト削減が可能となった。



（出典: Schuts, M. , Radboud University, Improving Software Development, 2010）

図 12-13:X 線 CT スキャンのアーキテクチャの概要

- 判断
  - 形式手法を利用した動機
    - 開発コストの削減
    - テストおよび統合フェーズにおける工数削減
  - 手法・ツール選択理由
    - 現場で利用することが容易な形式手法ツールであったため。
- 情報源
  - Robert C. Howe, ASD SaaS Business Case for Philips Healthcare

12.2.22. ClearSy - コンポーネント仕様のモデル化

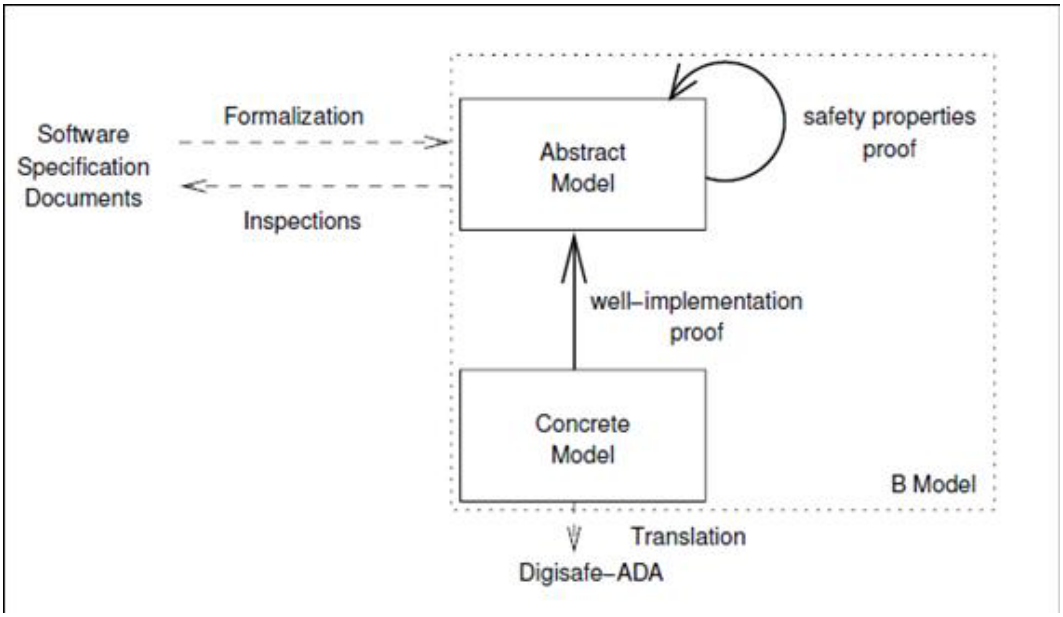
ドメイン	自動車
開発対象	車載コンポーネントの機能
国	フランス
開発組織	ClearSy
形式手法（言語、ツール）	Event B (Atelier B)
適用範囲・規模（形式手法）	• 一部の機能 • 50B モデル • 7000 イベント • 2000 抽象変数 • 1 台目は 14 人年、2 台目は 5.6 人年
適用対象のソフト種別	制御系
適用目的・工程	実機テスト
実装言語	不明
実装規模	不明
効果	自動車の各コンポーネントが、仕様に基づいて正しく作成されているかどうかが確認された

- 詳細情報
  - 検証内容
    - 自動車の全てのコンポーネントについて、正しくて完全な Event-B モデルを作成した
    - 実際の自動車のコンポーネントとその挙動と、Event-B モデルとの明確な対応付けを行った
    - コンポーネントの挙動を記録し、Event-B モデルと整合しているかどうかの確認を行った

- 判断
  - 形式手法を利用した動機
    - 従来の不具合発見手法では、複雑化した自動車の不具合を見つけることが困難になってきた
  - 手法・ツール選択理由
    - 仕様の正確な記述を行う必要があった
  - 障害と工夫
    - 自動車の全てのコンポーネントについて実機の検証を行うのはコストが大きいため、重要なシナリオのみを対象とした
- 情報源
  - Guilhem Pouzancre, How to diagnose a modern car with a formal B model, 2003

12.2.23. ClearSy - シャルルドゴール空港の無人シャトル制御

ドメイン	鉄道
開発対象	無人シャトル制御システム
国	フランス
開発組織	ADP (Paris Airport)、Siemens Transportation Systems (STS)、ClearSy
形式手法（言語、ツール）	B (EDiTh B, Bertille, Atelier B)
適用範囲・規模（形式手法）	ソフトウェア仕様ドキュメント: 228 ページ（84 の機能モジュール） B モデル: 183,897 行
適用対象のソフト種別	リアルタイム制御系
適用目的・工程	要求仕様、設計、コード生成
実装言語	Ada
実装規模	Ada: 158,612 行
効果	限られた予算、納期内で、安全要求を全て満たし、バグがほとんど無いシステムの開発を行えた 開発されたソフトウェアは IEC 61508: EN 50126, EN 50128, EN 50129 に準拠。また、SIL4 に分類された。



（出典: ClearSy and Siemens Transportation Systems, Using B as a High Level Programming Language in an Industrial Project: Roissy VAL, 2005）

図 12-14：無人シャトル制御システムにおける抽象モデルと具体モデルとの関係

- 詳細情報
  - 検証内容
    - 制御ユニットが、コントロールセンターから無人シャトルの経路についての命令を受け取り、それに基づいて制御ユニットが無人シャトルを制御するシステムにおいて、全ての安全要求を満たしていることを検証した
  - 検証規模
    - B モデルで記述した全ての要素間に矛盾が無いことを検証した
    - リファインメントした際に矛盾が生じていないことを検証した
- 判断
  - 形式手法を利用した動機
    - 限られた予算内で、初期リリースからほとんどバグの無い巨大なソフトウェア構築が求められた。特に安全要求への対応が求められた。
  - 手法・ツール選択理由
    - ソフトウェア仕様書がとても精度が低いものであり、システムを理解するためには何らかの形式化が必要であった。
  - 障害と工夫
    - 自動リファインメントを行うには問題の規模が大きすぎたため、手動で B モデルを分割した
    - 自動リファインメントを実施すると、実行速度が遅いアルゴリズムを採用してしまうことがあったため（本来であれば線形時間で解けるアルゴリズムを指数時間かかるアルゴリズムにしてしまった）、その場合は手動で新しいルールを作成し、適用した
    - 仕様書の理解を助けるために、質問/回答データベースを作成し、ClearSy の質問と Siemens の回答の対応付けを行った
    - 記述した B モデルが、自然言語で記述された仕様書を正しく表現していることを確認するために、全ての B モデルの要素について、B モデルを記述していないメンバーがチェックした
    - B モデルと自然言語のトレーサビリティを確保するために、B モデルの各オペレーシ

ョンにおいてコメントを記述した

- 組織
  - 教育
    - 仕様書を理解するためのドメイン知識及び B の読解能力、記述能力の教育を行った。詳細化のモデルに関してはツールの導入により負荷の軽減を図った。
- 情報源
  - ClearSy and Siemens Transportation Systems, Using B as a High Level Programming Language in an Industrial Project: Roissy VAL, 2005

12.2.24. POSCON - 地下鉄のプラットフォームドア

ドメイン	鉄道
開発対象	地下鉄のプラットフォームドア
国	ブラジル
開発組織	POSCON 社 (韓国 POSCO 社のグループ企業)
形式手法(言語、ツール)	SCADE
適用範囲・規模(形式手法)	不明
適用対象のソフト種別	リアルタイム制御系
適用目的・工程	設計、コード生成
実装言語	C
実装規模	10000 行
効果	• RAMS SIL-3 の達成 • 信頼性 99.95%、可用性 99.998% • システムの信頼性及び生産性の改善 • 製品価値の向上及び運用コストの削減 • プラットフォームドアシステムの安全性の保証 • ソースコードにおけるV&V活動の工数を大幅に短縮することができ、厳しい納期に間に合った

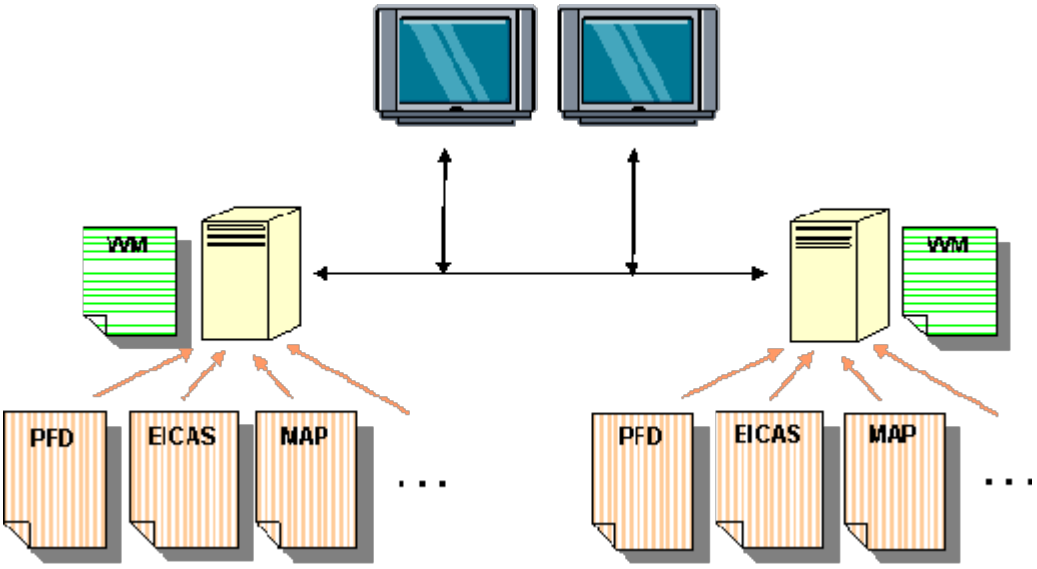
(出典: Esterel Technologies,「Poscon :: Success Stories」,
http://www.esterel-technologies.com/technology/success-stories/poscon)

図 12-15:開発されたプラットフォームドア

- 詳細情報
  - 検証内容
    - EN 50126 (IEC 62278)、EN 50128 (IEC 62279)、EN 50129 (IEC 61508)に基づく安全性の検証
- 判断
  - 形式手法を利用した動機
    - プロジェクト期間がとても短かった
    - SIL3 のためには、ソフトウェアのソースコードレベルを含む様々な V&V 活動が要求された
  - 手法・ツール選択理由
    - SCADE Suite は、SIL3-4 に対応した唯一の C 言語生成ツールであった
- 組織
  - 体制
    - POSCON 社の 3 名が開発を行った。開発者はソフトウェア工学の学位を持つが、形式手法については知識や経験は無かった。
  - 教育
    - SCADE Suite はとても学習が容易であり、1 週間のトレーニングによって設計者はモデリングを開始できた。
- 情報源
  - POSCON 社プレゼンテーション資料（SCADE User Group Conference 2009）

12.2.25. Rockwell Collins - パイロット用のディスプレイを管理するシステム

ドメイン	航空運輸
開発対象	パイロット用ディスプレイを管理するソフトウェア（ADGS-2100 Adaptive Display and Guidance System Windows manager)
国	米国
開発組織	Rockwell Collins
形式手法（言語、ツール）	NuSMV, Simulink
適用範囲・規模（形式手法）	Simulink : 16117 ブロック 4295 サブシステム NuSMV :9.8x109～1.5x1037
適用対象のソフト種別	リアルタイム制御系
適用目的・工程	設計
実装言語	不明
実装規模	不明
効果	563 性質に対してエラー98 件を発見した



（出典: Michael W. Whalen, John D. Innis, Steven P. Miller, and Lucas G. Wagner, ADGS-2100 Adaptive Display & Guidance System Window Manager Analysis, NASA Contractor Report CR-2006-213952, 2006）

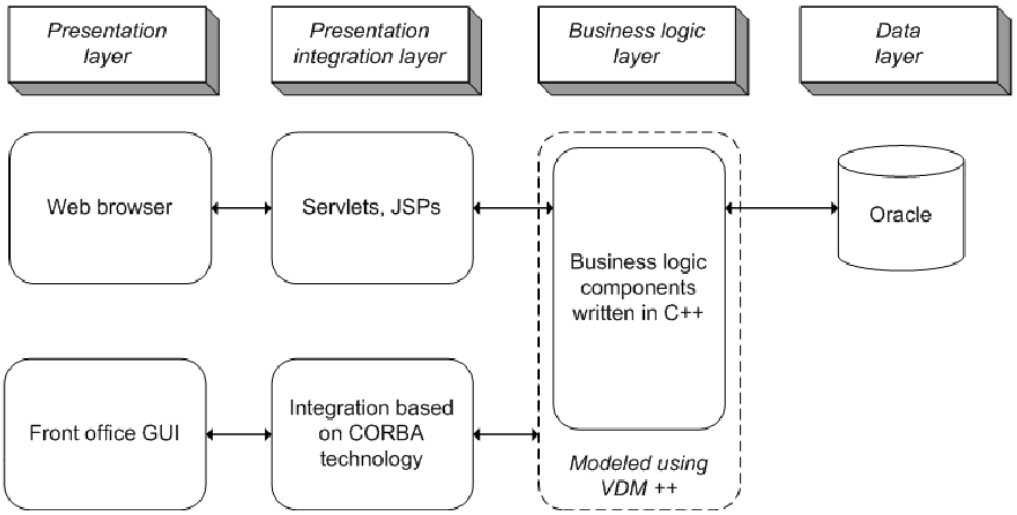
図 12-16：ディスプレイアーキテクチャ（オレンジ色はディスプレイアプリケーション、緑色はウィンドウマネージャ）

- 詳細情報
  - 検証内容
    - 異なるデータ表示アプリケーションから、適切なディスプレイにデータがルーティングする、ウィンドウマネージャの性質を検証する。
- 判断
  - 障害と工夫
    - 要求仕様を検証する性質に落とし込むこと
    - Simulink などの商用モデリングツールのデータを異なる種類の形式手法を用いた分析ツールの入力とする、変換ツールを作ること
    - 分析結果を分析者と製品エンジニアから分かりやすくするツールを作ること
    - アプリケーションを個々に分析できるようなサブシステムに分解する手法を策定すること
    - プロセスを改善できるように、検証プロセスを反復していくこと
    - Rockwell Collins 社とミネソタ大が共同開発した変換フレームワークを利用している。このフレームワークにより、商用モデリングツールの言語から、モデル検査器や定理証明器への変換ができるようになっている。
    - 次の 3 つの要件を満たす検証プロセスを策定すること
      - 分析結果の妥当性
      - ソフトウェア要求仕様からすべての形式的な検証性質が追跡可能であること
      - すべての要求項目が少なくとも一つの形式的な検証性質により検証されること
- 組織
  - 教育
    - 開発者は1日程度の訓練と作業中の助言だけでモデル検査ツールの実践的な導入が可能

・ 情報源
  - Steven P. Miller, Michael W. Whalen, Darren D. Cofer, Software Model Checking Takes Off, Communications of the ACM, February 1, 2010, pp.58-84.
  - NASA LaRC Formal Methods Program Research

12.2.26. CSK - TradeOne

ドメイン	バックオフィスシステム
開発対象	証券バックオフィスシステム TradeOne
国	日本
開発組織	CSK
形式手法（言語、ツール）	VDM++ (VDMTools)
適用範囲・規模（形式手法）	TradeOne 全体のうち、マル優サブシステムとオプションサブシステムに適用した（ソースコード行数の割合では約 6 割）。 マル優サブシステムに関して VDM++の行数は、ビジネスロジックが 8,102 行、制約が 1,539 行、TradeONE システムのユーティリティが 1,342 行、汎用的な目的のためのユーティリティが 774 行 合計 11,757 行。 オプションサブシステムに関しては、ビジネスロジックが 10,846 行、テストシナリオが 13,771 行、テストケースが 31,641 行等、合計 68,170 行
適用対象のソフト種別	エンタプライズ系
適用目的・工程	仕様記述、設計
実装言語	C++, Java
実装規模	TradeOne 全体では、1,342,858 DSI (DSI はソースコード行数とほぼ同義。COCOMO で利用される。) マル優サブシステムは 18,431DSI オプションサブシステムは 60,206DSI
効果	エラー発生率が、 • マル優サブシステムは 0 件 • オプションサブシステムは 0.05/KDSI • TradeOne 非適用部分では、0.67/KDSI であった。 生産性が、見積りよりも 50-60%程度向上した。



（出典: John Fitzgerald, Peter Gorm Larsen, Paul Mukherjee, Nico Plat, Marcel Verhoef, Validated Designs for Object-oriented Systems, version 1.2, 2004）

図 12−17：TradeOne システムの構成

- 詳細情報
  - 検証規模
    - テストの達成率は、C0（命令網羅）レベルで 98%であった。
  - 期間
    - 2000-2001
- 判断
  - 形式手法を利用した動機
    - 上流工程での検証が可能であるため、高信頼性、高生産性を実現できる。
- 組織
  - 体制
    - マル優サブシステム
      - 6 人チーム（ひと月あたり平均 4 人、3.5 か月間）で開発された。まずシステムアーキテクトが VDM++モデルの全体的な基本的なフレームワークを設計した。次に、4 人の VDM++経験者が、並行してモデルを作成した。全ての VDM++モデルはチーム全てのメンバーによってレビューされた。最後にフタルのプログラマが VDM++モデルから手動で実装した。
      - 全ての開発者は40歳を越えており、ソフトウェア開発経験は20年以上あった。
      - 4 人は、形式手法について基本的な事前知識があった。
    - オプションサブシステム：
      - 10 人チームで開発された。
      - まずドメインのエキスパートが日本語でオプションサブシステムの機能要求を記述した。
      - 並行して、マル優サブシステム開発者の VDM++エキスパートの一人が、1 週間以内で VDM++について指導した。
      - 次に、ドメインエキスパートがオブジェクト指向のエキスパートの協力を得て、UML でユースケースを書いた。
      - 並行して、マル優サブシステム開発者であったシステムアーキテクトが、VDM++モデルの全体的なフレームワークを記述した。
      - 次に、二人の VDM++エキスパートと、新しく指導された 3 人のプログラマが

VDM++モデルを完成させた。

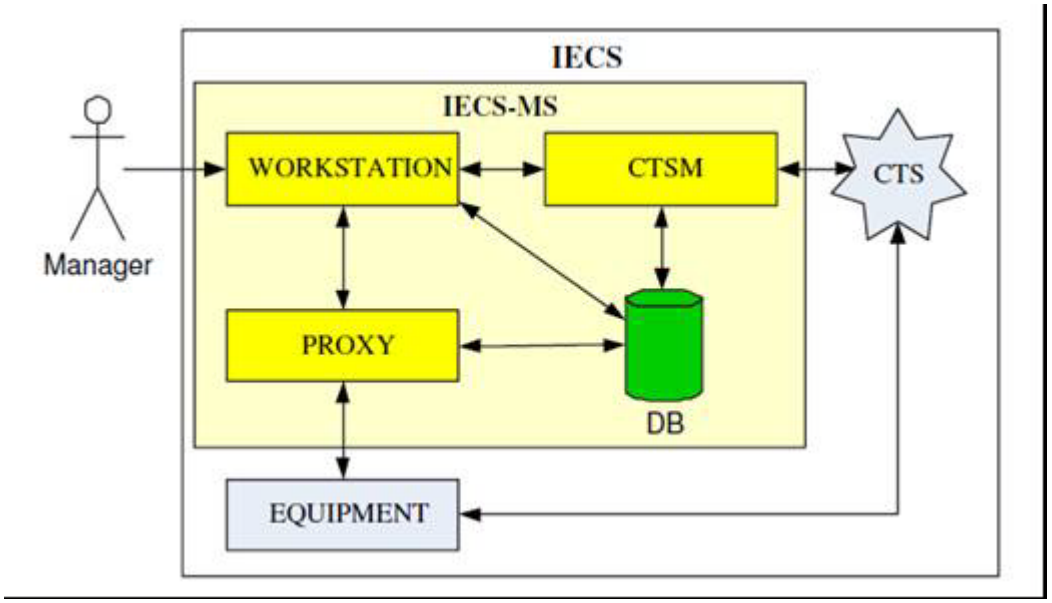
- 教育
  - ✧ オプションサブシステムの開発工数は合計で 60.1 人月であり、教育に要した時間はそのうちの 2.8 人月であった。

・ 情報源

- ソフトウェアコンポーネント技術に関する調査研究報告書, http://it.jeita.or.jp/eltech/report/2004/04-jou-4.html, JEITA
- John Fitzgerald, Peter Gorm Larsen, Paul Mukherjee, Nico Plat, Marcel Verhoef, Validated Designs for Object-oriented Systems, version 1.2, 2004, http://overtureeditor.googlecode.com/svn/trunk/Documentation/literature/VDM%20Book/bookfinal.pdf
- VDM information web site - 導入事例, http://www.vdmtools.jp/modules/tinyd1/index.php?id=5

12.2.27. Selex Communications - 船舶通信システム

ドメイン	船舶
開発対象	船舶通信システム IECS
国	イタリア
開発組織	Selex Communications
形式手法（言語、ツール）	SPIN (ステート図や、PSC: Property Sequence Charts に基づく検証プロパティを CHARMY で記述し、SPIN 用に変換。)
適用範囲・規模（形式手法）	不明
適用対象のソフト種別	制御
適用目的・工程	仕様記述、設計
実装言語	不明
実装規模	不明
効果	各プロパティの検証および、デッドロックや到達不能パスや正しくない最終状態が存在しないことを検証できた。



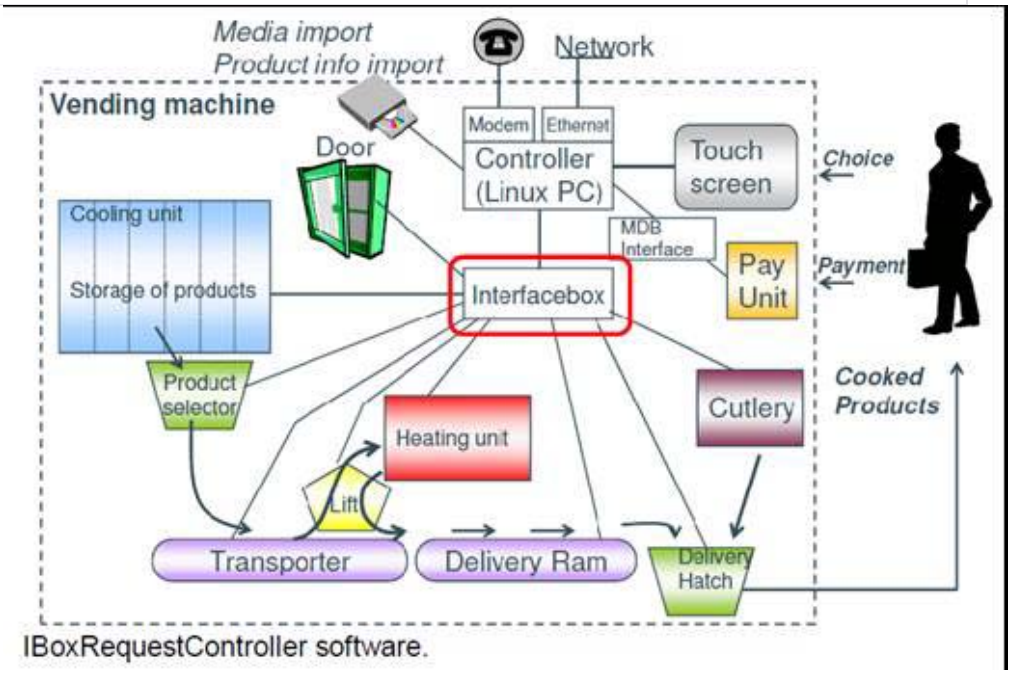
（出典: Daniela Colangelo, Daniele Compare, Paola Inverardi, and Patrizio Pelliccione, Reducing Software Architecture Models Complexity: a Slicing and Abstraction Approach, Formal Techniques for Networked and Distributed Systems-FORTE 2006）

図 12-18:IECS ソフトウェアの構成図

- 詳細情報
  - 検証内容
    - デッドロックが無いこと、正しくない最終状態が存在しないこと、到達不能なパスが存在しないこと。その他、Activate Service (ユーザがサービスを非活性化させていないとき、ActivateService リクエストの送信後、サービスがアクティベートされる)、Deactivate Service、Reconfiguration Service、Modify Equipment、Modify Equipment by Service の各プロパティ。
  - 検証規模
    - SPIN でのデッドロックフリー等の検証時のモデルのサイズは以下のとおりであった。
      - 状態数: 1.3e+08
      - 遷移数: 6.2e+08
      - メモリ使用量: 2.0Gb
    - 各プロパティの検証については、状態爆発を起こしてしまったため、スライシング技術を利用してモデルサイズを削減し、検証を実施した。
- 判断
  - 障害と工夫
    - モデルのサイズを小さくするため、CARMY で記述されたソフトウェアアーキテクチャモデルから、TESTOR を拡張して構築した DEPCOL アルゴリズムやスライシング技術によって検証すべき部分を抽出した。
- 情報源
  - Daniela Colangelo, Daniele Compare, Paola Inverardi, and Patrizio Pelliccione, Reducing Software Architecture Models Complexity: a Slicing and Abstraction Approach, Formal Techniques for Networked and Distributed Systems-FORTE 2006

12.2.28. Sioux - ホットフード自動販売機

ドメイン	その他
開発対象	ホットフード自動販売機
国	オランダ
開発組織	Sioux Embedded Systems
形式手法(言語、ツール)	Verum ASD Suite
適用範囲・規模(形式手法)	自動販売機の中心となるコンポーネント
適用対象のソフト種別	制御系
適用目的・工程	設計、自動コード生成
実装言語	C#
実装規模	2,889LOC
効果	品質および生産性が向上した。(ASD の教育時間を除くと 30%生産性が向上した。教育時間を含めると 1%生産性は低下した。)



(出典: Leon Bouwmeester and Arjen Klomp, Improving productivity and quality using the ASD:Suite, Sioux evaluates Verum's ASD for embedded software, 2009)

図 12-19:開発された自動販売機の構成図

- 詳細情報
  - 検証内容
    - デッドロック、ライブロック、レースコンディションの有無を検証した。
  - 期間
    - 2 か月間
- 判断
  - 形式手法を利用した動機
    - 形式手法を利用することで、従来の開発よりも生産性や品質に良い影響を与えるか

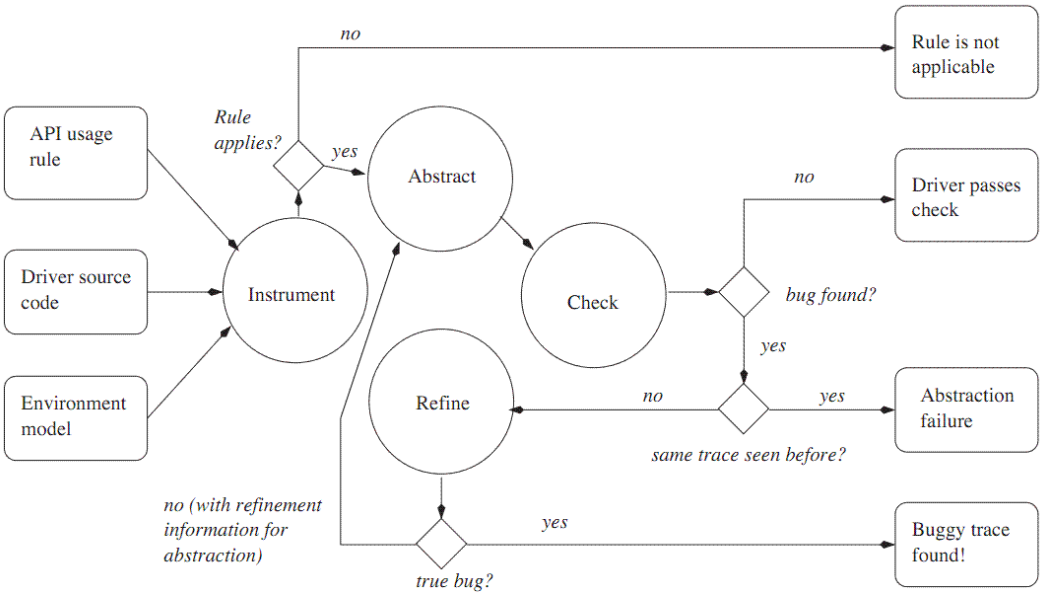
どうかの評価を行うためであった。（該当ソフトウェアは従来手法で既に開発済みであり、今回新たに ASD を用いて再開発を行い、比較した）

- 組織
  - 体制
    - 熟練のソフトウェアエンジニアがフルタイム、熟練のソフトウェアアーキテクトが 20% のエフォートで開発に携わった。
  - 教育
    - ASD の教育に 2 人合計で 140 時間を費やした。
  - その他
    - Verum のアシスタントが必要に応じてサポートした。

- 情報源
  - Leon Bouwmeester and Arjen Klomp, Improving productivity and quality using the ASD:Suite, Sioux evaluates Verum's ASD for embedded software, 2009

12.2.29. Microsoft - Windows デバイスドライバ

ドメイン	その他
開発対象	126 の WDM (Windows Deriver Model) ドライバと 20 の KMDF (Kernel Mode Driver Framework) ドライバの検証
国	米国
開発組織	Microsoft
形式手法（言語、ツール）	SLAM (Static Driver Verifier tool)
適用範囲・規模（形式手法）	WDM ドライバについては 60 のルール（検証項目）を設定し、KMDF ドライバについては 40 のルールを設定した。
適用対象のソフト種別	リアルタイム制御系
適用目的・工程	コード検証
実装言語	C
実装規模	48-130,000LOC であり、平均 12,000LOC
効果	長年利用されてきたドライバの不具合を多数発見することができた。



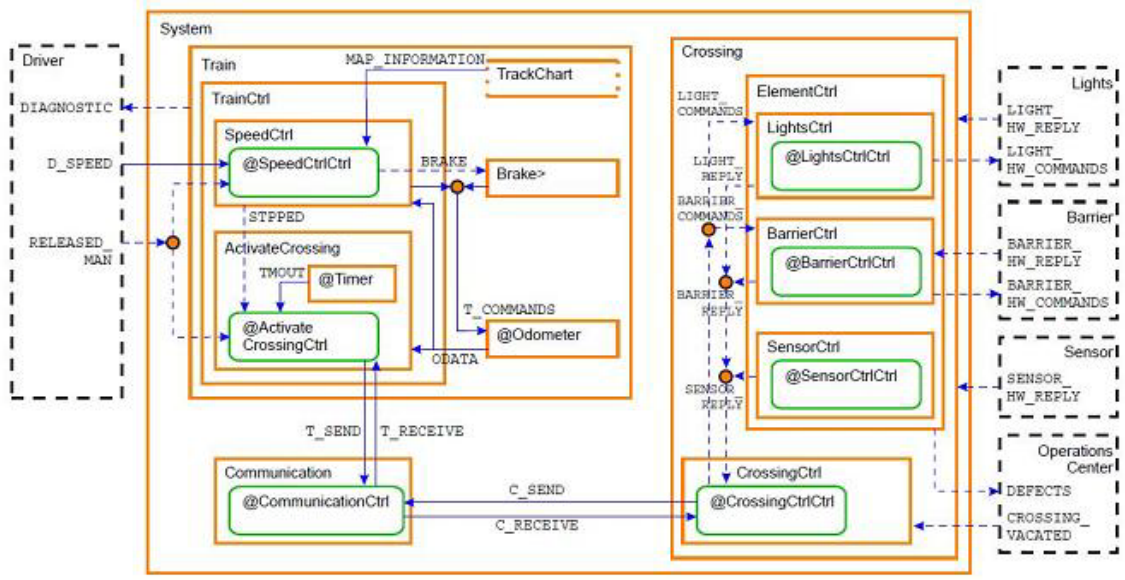
（出典: Ball, T. and Bounimova, E. and Cook, B. and Levin, V. and Lichtenberg, J. and McGarvey, C. and Ondrusek, B. and Rajamani, S.K. and Ustuner, A., Thorough static analysis of device drivers, Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006）

図 12-20：Static Driver Verifier tool の解析エンジンアーキテクチャ

- 詳細情報
  - 検証内容
    - IoCompleteRequest が呼び出されているときはドライバは STATUS_PENDING を返さない
    - もし存在するならプラグアンドプレイ I/O リクエストパケットはより低レイヤのドライバに渡される
    - IoAttachDeviceToDeviceStack は適切なデバイスオブジェクトと共に呼び出される等。
  - 検証規模
    - 約 93%について自動的に検証することができた
- 判断
  - 形式手法を利用した動機
    - コードが取りうる全ての振舞いを検証するため
  - 手法・ツールの選択理由
    - Windows のデバイスドライバを検証するため
    - 検証のための入力を必要としないため
  - 障害と工夫
    - SDV における Windows カーネルモデルやルール（検証項目）の不具合等により、デバイスドライバに不具合が無くても不具合があると報告されることがあったため、これらの修正を並行して行った。
- 情報源
  - Ball, T. and Bounimova, E. and Cook, B. and Levin, V. and Lichtenberg, J. and McGarvey, C. and Ondrusek, B. and Rajamani, S.K. and Ustuner, A., Thorough static analysis of device drivers, Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006

12.2.30. ボンバルディア - 鉄道制御システム

ドメイン	鉄道
開発対象	Deutsche Bahn 社の無線鉄道制御システム
国	ドイツ
開発組織	ボンバルディア
形式手法（言語、ツール）	Rational Statemate/LSC (Live Sequence Chart)
適用範囲・規模（形式手法）	不明
適用対象のソフト種別	リアルタイム制御系
適用目的・工程	要件定義、システム設計、ソフトウェア設計
実装言語	不明
実装規模	不明
効果	形式的に要求を記述することにより、仕様の厳密さが高まり、再利用性も向上した。



（出典: Brill, M. and Buschermohle, R. and Damm, W. and Klose, J. and Westphal, B. and Wittke, H., Formal verification of LSCs in the development process, Integration of Software Specification Techniques forApplications in Engineering, pp.494--516, 2004）

図 12-21：鉄道制御システムの主要な Statemate モデル

- 詳細情報
  - 検証内容
    - 列車が地上子などのアクティベーションポイントに近づいたら必ず通信チャネルがセットアップされる。
- 判断
  - 形式手法を利用した動機
    - 開発するコンポーネントが安全要求を満たしていることを保証するため。
    - モデル検査はテストと異なり全ての入力および入力の組合せについて検査すること

ができるため。

- 手法・ツール選択理由
  - Statemate に組み込まれている Live Sequence Charts はよく知られている MSC2000 の Message Sequence Chart や UML のシーケンス図を基にしており、クリティカルな通信プロトコルの記述および検証に向いているため。
- 障害と工夫
  - モデル検査はモデルを解析するのみであり、ハードウェアやソフトウェアの実装に関して検証するものではない。そのため、作成したモデルから自動的にテストを作成し、実装に適用した。

・ 情報源

- Bohn, J. and Damm, W. and Klose, J. and Moik, A. and Wittke, H. and Ehrig, H. and Kramer, B. and Ertas, A., Modeling and validating train system applications using statemate and live sequence charts, IDPT, 2002
- Brill, M. and Buschermohle, R. and Damm, W. and Klose, J. and Westphal, B. and Wittke, H., Formal verification of LSCs in the development process, Integration of Software Specification Techniques forApplications in Engineering, pp.494--516, 2004