

SystemCベースのTLMと 動作合成に関する取り組み

2007年1月26日

JEITA EDA技術専門委員会

標準化小委員会

SystemCタスクグループ



目次

- JEITA SystemC-TGについて
- TLM動向調査
- SystemC Synthesizable Subset レビュー
- SystemC 動作合成スタイルガイド構成要件
- 発表まとめ
- 付録・資料編

- 
-
- JEITA SystemC-TGについて
 - TLM動向調査
 - SystemC Synthesizable Subset レビュー
 - SystemC 動作合成スタイルガイド構成要件
 - 発表まとめ
 - 付録・資料編



SystemCタスクグループについて

■ 設立の背景

- 2003年10月に、JEITA EDA技術専門委員会 標準化小委員会内に設置(SystemVerilogタスクグループと同時)
- SystemCが、SoC(System on Chip)の開発のためのシステムレベル記述言語のひとつとして幅広く使われてきており、かつ正式な標準化の機運が高まってきた

■ 目的

- 日本国内における唯一のSystemCの標準化関連組織として、OSCIやIEEE P1666ワーキンググループと連携しつつ、日本国内の事情・要求事項を取り込むべくSystemCの国際標準化を進めていく。
- SystemCに関連した調査結果を積極的に情報発信を行うことで、国内普及を図る。これらにより日本の産業界の国際競争力を高めることを目指す。



SystemCタスクグループメンバー

主査	長谷川 隆	(富士通)	
副主査	今井 浩史	(東芝)	
委員	中西 早苗	(NECエレクトロニクス)	
	清水 靖介	(沖電気)	
	逢坂 孝司	(ケイデンス)	
	長尾 文昭	(三洋半導体)	
	中野 淳二	(シノプシス)	
	山田 晃久	(シャープ)	
	柿本 勝	(ソニー)	
	竹村 和祥	(松下電器)	
	菊谷 誠	(メンター)	
	里井 朋樹	(リコー)	
	渡邊 政志	(ルネサステクノロジ)	
	客員	今井 正治	(大阪大学)

(計14名、2007年1月26日現在)



活動内容と主な成果

■ SystemC標準化活動

- IEEE P1666 WGに投票権のあるメンバーとして参加し、2005年12月に正式承認されるまでSystemCの言語仕様書のレビューを実施し、50件以上の改善提案を行い、採択された(2005年度まで)

■ SystemC技術調査

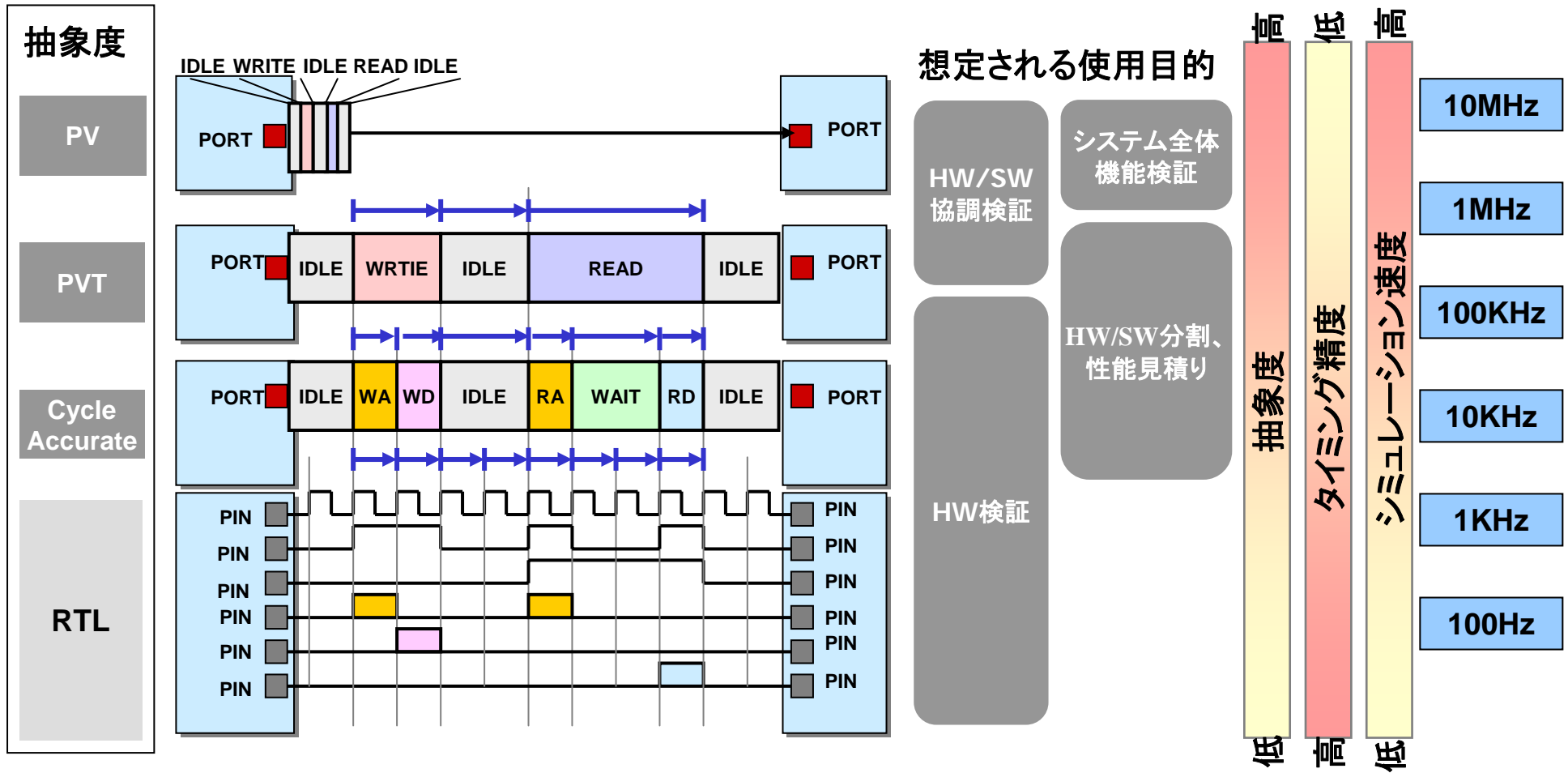
- 過去5年間における国内外でのSystemCの利用状況について調査を実施(2005年度まで)
- SystemC 2.1について調査を行い、その特長を日本語で紹介(2005年度)
- 国内外におけるTLM(Transaction Level Modeling)の利用状況調査を実施(2006年度)
- OSCIより公開されている合成サブセットのドキュメントのレビュー及び抄訳の作成を実施(2006年度)

■ SystemC普及活動

- SystemCユーザフォーラムを開催し、積極的に情報発信を行いSystemCを利用した設計の普及をはかる
- SystemC動作合成ガイドライン構成要件を検討中(2006年度)

- 
-
- JEITA SystemC-TGについて
 - **TLM動向調査**
 - SystemC Synthesizable Subset レビュー
 - SystemC 動作合成スタイルガイド構成要件
 - 発表まとめ
 - 付録・資料編

TLMの抽象度定義と想定される使用目的





国内外各社TLM利用状況調査

- 国内・国外のSystemCユーザーがTLMをどの抽象度(PV、PVT、CA)で何を目的に使用しているかを調査。
- 調査対象
 - 国内: JEITA SystemC TG参加企業
 - 国外: ECSI Institute Workshop on Users Experience with TLM, June 8, 2006 - Paris, France
- TLM使用目的の分類
 - システム全体機能検証
 - HW/SW分割、性能見積り
 - HW/SW分割、アーキテクチャ検討・生成、性能見積り
 - HW検証
 - HWモジュールの検証、例えば、RTL、高位合成用記述をTLMテストベンチで検証
 - HW/SW協調検証
 - HWモデル上でのデバイスドライバ、アプリケーションSWの設計・検証

国内外各社TLM利用状況調査(結果)

使用目的 抽象度	想定				国内				国外(ヨーロッパ)				
	システム全体機能検証	HW/SW協調検証	HW/SW分割、性能見積り	HW検証	システム全体機能検証	HW/SW協調検証	HW/SW分割、性能見積り	HW検証	システム全体機能検証	HW/SW協調検証	HW/SW分割、性能見積り	HW検証	
PV	●	●				●	○			○	●	●	●
PVT		●	●	●	●	○	●		○	●	●	●	●
CA			●	●	●	○	●	●		●	●		

国内: OSCIL TLM1.0をそのまま使ったケース無し。独自API。

○: C/C++使用、●: SystemC使用

■ CALレベル主体の低抽象度のTLMの使用

- CAモデルはシミュレーション速度が不十分。生産性低い。
- 生産性向上のために、より高抽象度のTLM使用が必要。
- しかし、現状PV、PVTモデル利用のメソッドの確立が不十分で高抽象度への移行ができていない。

■ 独自APIの多用

- OSCI TLM1.0をそのまま使った例は無し。独自API多数。
- モデルの再利用性への障害。



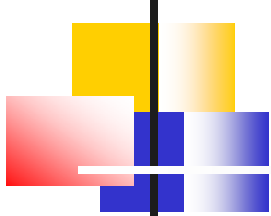
低スピード
低流通性
高コスト

解決策はPV、PVTのメソッドの確立、TLM APIの標準化！



TLMメソドロジ成功の鍵

- PV、PVTのメソドロジ確立・普及のために
 - 成功(失敗)事例、適用事例などの積極的な情報発信
 - ユーザフォーラムの開催
- TLM APIの標準化への寄与
 - OSCI TLM2.0調査とフィードバック
 - OSCI TLM2.0でPV/PVTのレベルは共有可能になることを期待。(OSCI TLM1.0では低抽象度トランザクションレベルの基本I/Fのみ標準化。もっと高抽象度の標準が欲しい。)

- 
-
- JEITA SystemC-TGについて
 - TLM動向調査
 - SystemC Synthesizable Subset レビュー
 - SystemC 動作合成スタイルガイド構成要件
 - 発表まとめ
 - 付録・資料編

最終版ではありませんが、ドラフトの抄訳を参考資料
「OSCI-SystemC合成サブセットDraft1.1.18の概要」として添付しています



SystemC Synthesizable Subsetの概要

- OSCI Synthesis WGが策定した合成技術対応のSystemCサブセット
 - V1.1.18は2004年12月に作成
 - OSCIより2006年6月に一般レビュー公開されたので、SystemC-TGで内容をレビュー ⇒ OSCIへ57件を提言
- 基本構成
 - C++のISO規格とSystemC 2.0.1に基づいている
 - 論理合成と動作合成のためのサブセットを規定
 - C++の構文(BNF)で扱い方を規定
 - **Supported:** サポートする
 - **Extended:** 言語を拡張
 - **Ignored:** 無視する
 - **Not Supported:** サポートしない



SystemCの合成可能な構文

- `sc_module`クラスのモジュール定義および階層記述
- `SC_METHOD`プロセスと`SC_CTHREAD`プロセス
- `sc_in`, `sc_out`を用いた入出力と`sc_signal`の内部信号
- `sc_fix`, `sc_ufix` (ビット幅等が可変)を除くSystemCのデータ型

モジュールを表現するための構文が追加された



C++の使用可能な構文

- `wchar_t`を除く組み込み整数型とその演算
- 関数の利用
- すべての分岐文とループ文
- ネームスペース
- `union`を除くクラスの構文
- 関数や演算子のオーバーロード
- テンプレート
- プリプロセッサ

C++の特徴となっているほとんどの構文が利用できる

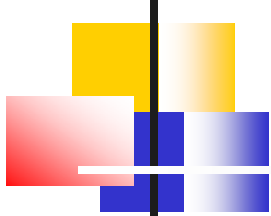


C++の使用できない構文

- double/floatの組み込み浮動小数点型
- 文字と文字列のリテラル
- ポインタ
- union
- 例外処理 (try, catch, throw)

<<OSCIへの提言の一例>>

文字と文字列のリテラルや静的なメモリをアクセスを利用するポインタは、使用可能にすべきである

- 
-
- JEITA SystemC-TGについて
 - TLM動向調査
 - SystemC Synthesizable Subset レビュー
 - SystemC 動作合成スタイルガイド構成要件
 - 発表まとめ
 - 付録・資料編



動作合成スタイルガイドの必要性

■ 動作合成スタイルガイドの目的

- 高位設計技術の活用による生産性向上
 - 共通スタイルにより動作合成の性能向上を期待
 - 動作合成を使った設計フロー構築を容易にする
- 設計資産の再利用による設計効率向上
 - 設計方法や記述スタイルの共通化で動作合成の実用性向上
 - 動作合成向け記述の共通化で再利用性を高める

■ 動作合成スタイルガイドの要件を作成する目的

- 高位設計による設計効率化推進として、動作合成の利用技術の共通化を目指す
- 動作合成を活用する基本技術を共有化
- ユーザの視点でスタイルガイドで規定すべき項目を規定
- ベンダ固有のノウハウから脱却して、共通資産蓄積を推進
- 開発側、ユーザー側に基本的なスタイルを提案



スタイルガイドの対象者とスコープ

- ハードウェア設計者でSystemCの初心者やSystemCを使っているが動作合成は初心者という人を主な対象者としている
- 動作合成の基本的な機能とその使い方を説明する
- SystemCによる高位設計フローの概説と機能検証についても盛り込む
- 動作合成については、ベンダー非依存で必要となる基本的な回路構成方法や合成に対する指示や合成結果の検証方法を示す
 - 動作合成を利用するための機能分割やSystemCを使った記述方法を説明
 - 動作合成を意識したテストベンチの作成方法と合成前後の機能検証方法を説明



動作合成スタイルガイドの構成

幅広く受け入れられるように大きく分けて次の3つのパートで構成する。

■ 導入

マネージャーや動作合成を未経験の設計者に動作合成の仕組みと効果を説明し、動作合成ツールと設計者の役割分担を正しく認識してもらう

■ 初級編

動作合成を用いた一通りの設計作業ができるレベルまで、SystemC記述の側面から説明する

■ 上級編

合成サブセットの詳細、再利用性や合成後の性能を向上させる記述スタイル、ツール依存の記述スタイルなどを説明する



動作合成スタイルガイドの章立て

章立ては以下の順および項目であることを推奨

- 導入

- 第1章 概論

- 第2章 動作合成とは

- 初級編

- 第3章 動作合成のための記述スタイル

- 第4章 検証環境

- 上級編

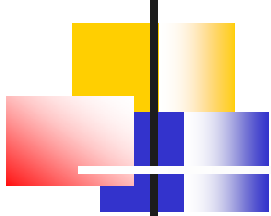
- 第5章 SoCへの実装

- 第6章 RTLとの対応

- 第7章 上級テクニック

- 第8章 ベンダ依存

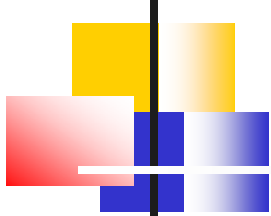
- 第9章 SystemCの合成サブセット

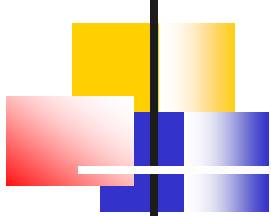
- 
-
- JEITA SystemC-TGについて
 - TLM動向調査
 - SystemC Synthesizable Subset レビュー
 - SystemC 動作合成スタイルガイド構成要件
 - **発表まとめ**
 - 付録・資料編



まとめ

- 標準化されたSystemCの活用に向けて
- TLM現状の課題
 - 低い抽象度(CA)、独自APIで利用が大半のため、シミュレーションスピードとモデルの再利用性がまだ低い
- TLMメソッドロジ成功の鍵
 - TLM APIの標準化、OSCI TLM2.0へ期待
- OSCI合成サブセットに57件のフィードバック
 - 今後の標準化に向けて貢献
- 動作合成スタイルガイド構成要件
 - 動作合成活用の指針を提案、ツールが対応することを期待

- 
-
- JEITA SystemC-TGについて
 - TLM動向調査
 - SystemC Synthesizable Subset レビュー
 - SystemC 動作合成スタイルガイド構成要件
 - 発表まとめ
 - 付録・資料編
 - 資料1: SystemCユーザ・フォーラム・アンケート結果
 - 資料2: OSCI-SystemC合成サブセットDraft 1.1.18の概要



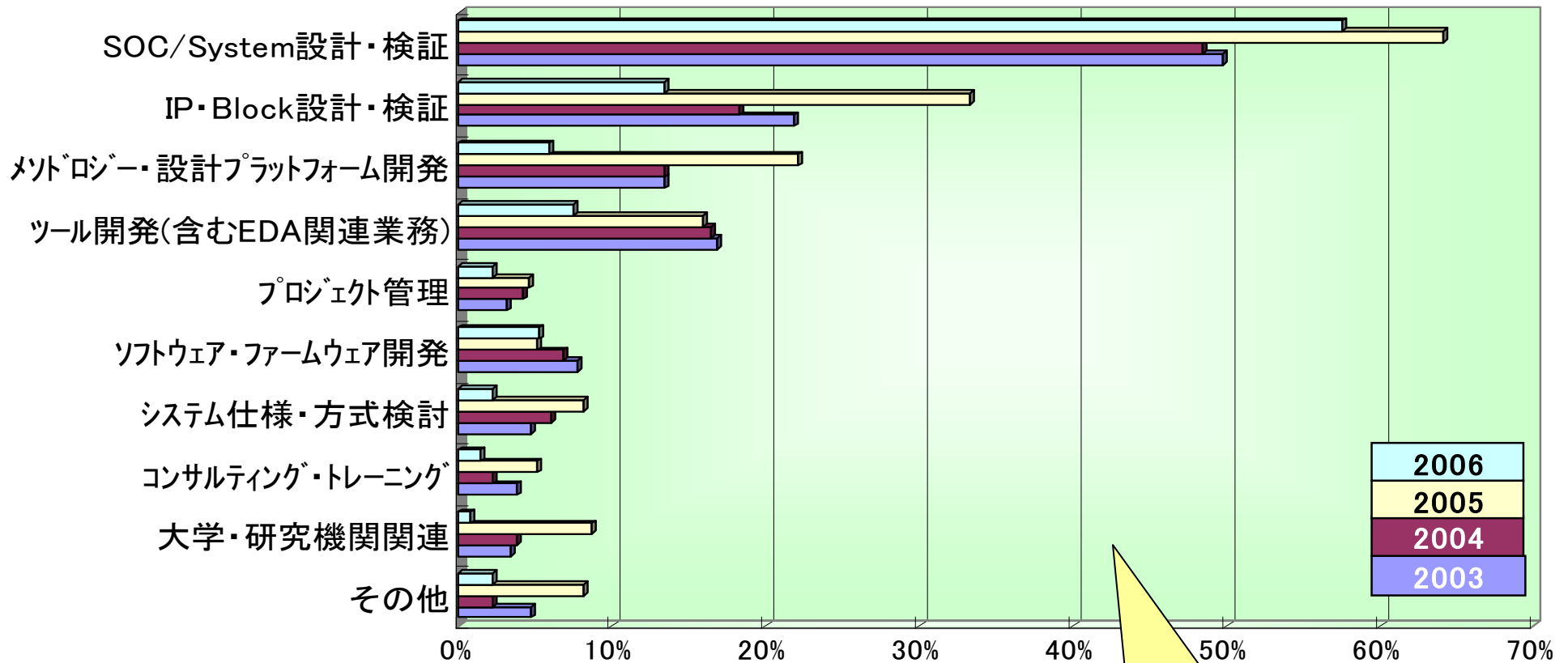
資料編 1. SystemC ユーザフォーラム・アンケート 調査結果(2003～2006)



アンケート調査結果(2003～2006)

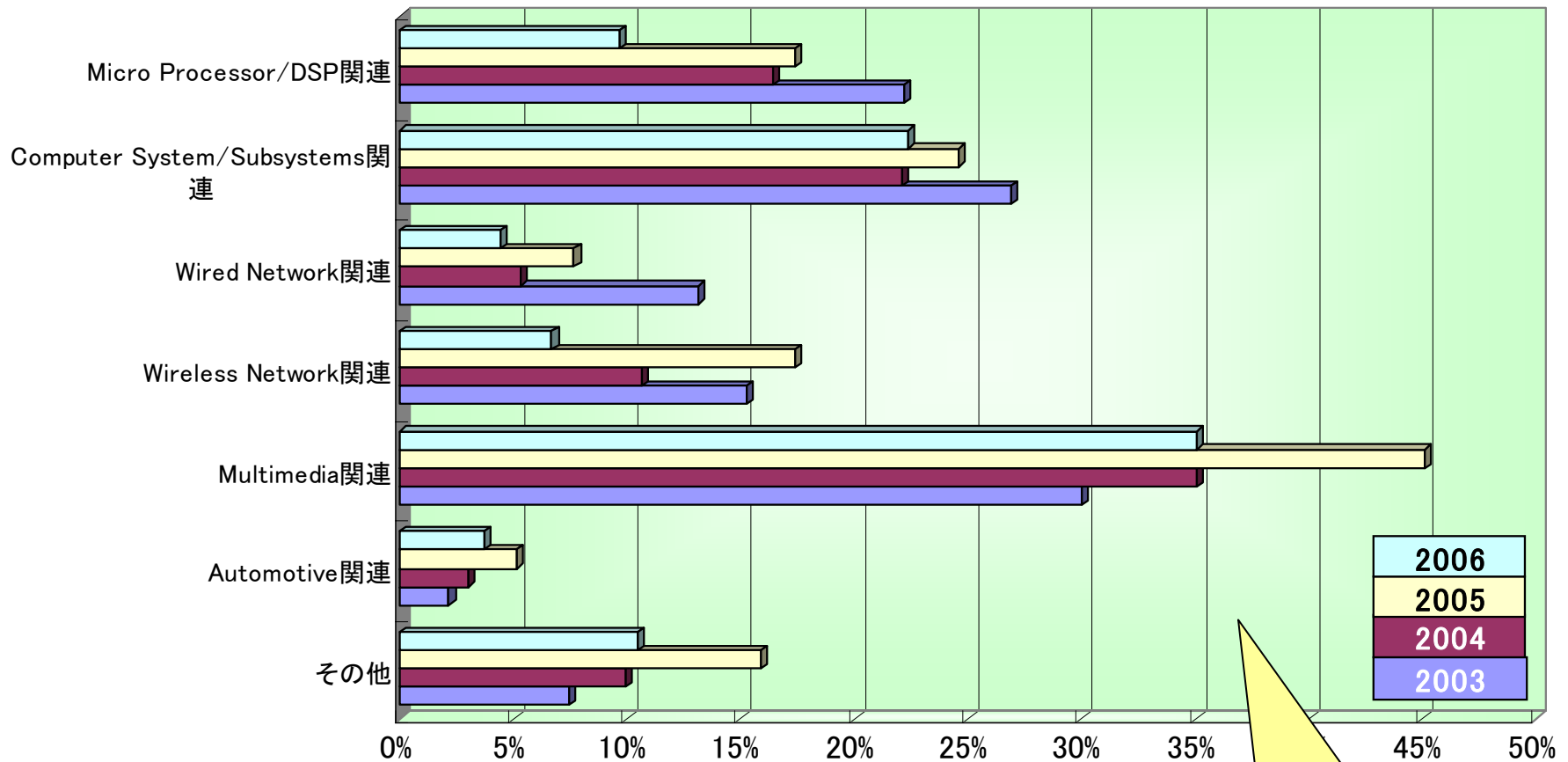
- SystemCユーザフォーラム2003,2004(OSCI主催)、同2005,2006(JEITA EDA技術専門委員会主催)のアンケート調査を集計し、聴講者の動向分析を行った
 - 参加者のビジネス分野、アプリケーション分野はあまり変化はみられない。(1),(2)
 - 発表内容では、ユーザ事例に関する関心が高くなってきている。SystemCの言語仕様や概念といった初心者的な内容より、より具体的な使い方といった所へ興味がシフトしてきている。(3)
 - 高位(動作)合成の興味が高い。他国の学会の発表事例数等と比較すると、日本の特殊性が表れている。(4)(5)
 - 等価性検証・プロパティチェックの興味が伸びている。これは高位(動作)合成を取り入れた設計もかなり浸透してきており、その次の問題と考えている参加者が多いものと考えられる。(6)
 - 標準化・言語拡張では、TLMの標準化を望む声が多くなり、逆にHW/SW間I/Fへの期待が少なくなっている。SystemCを使う設計者が、よりHW設計者の方に多くなってきているものと考えられる。(7)(8)

1. ご担当業務またはビジネスは?



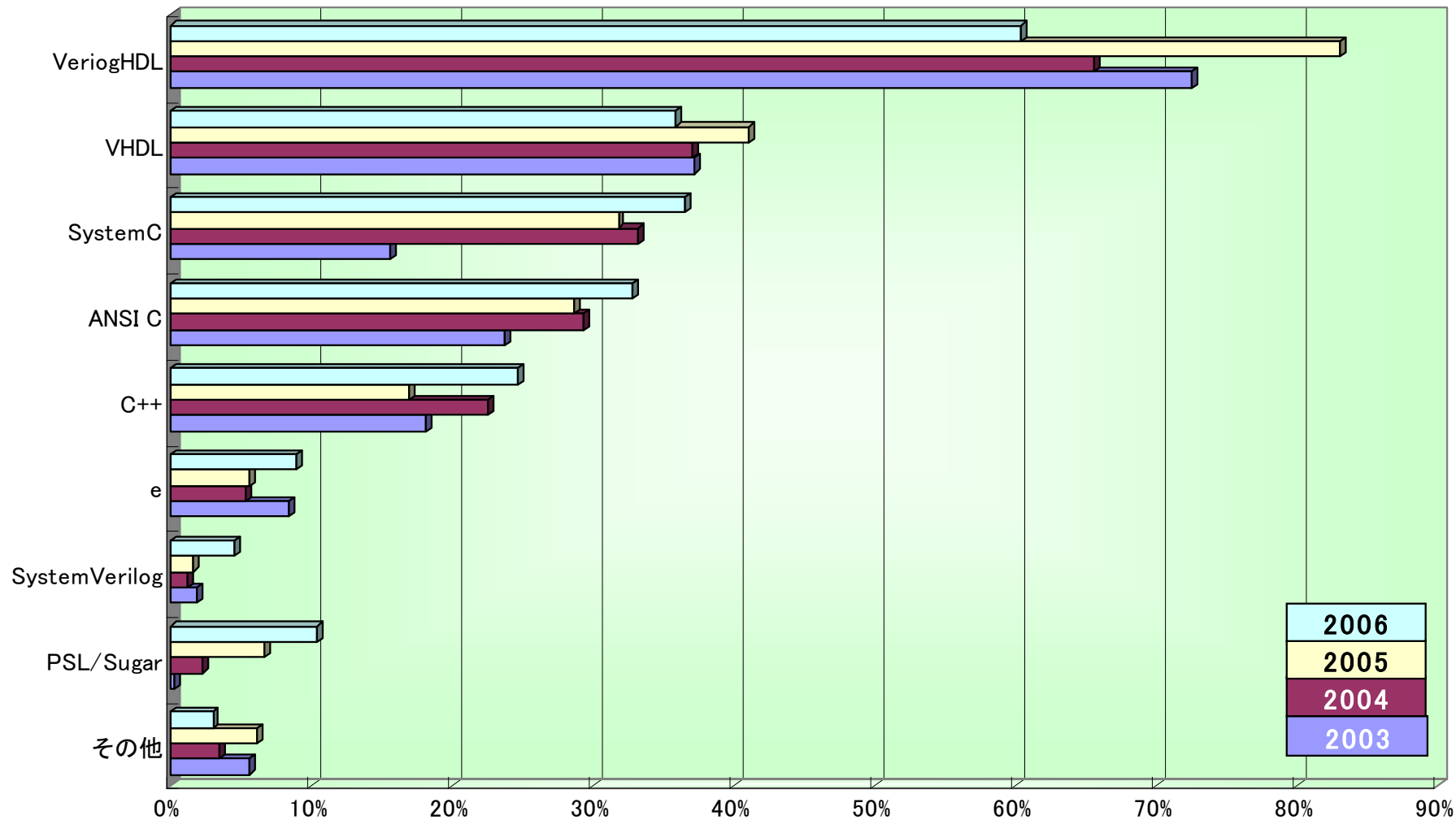
(1) 年度によって構成の大きな変化なし

2. ご担当製品アプリケーションは？

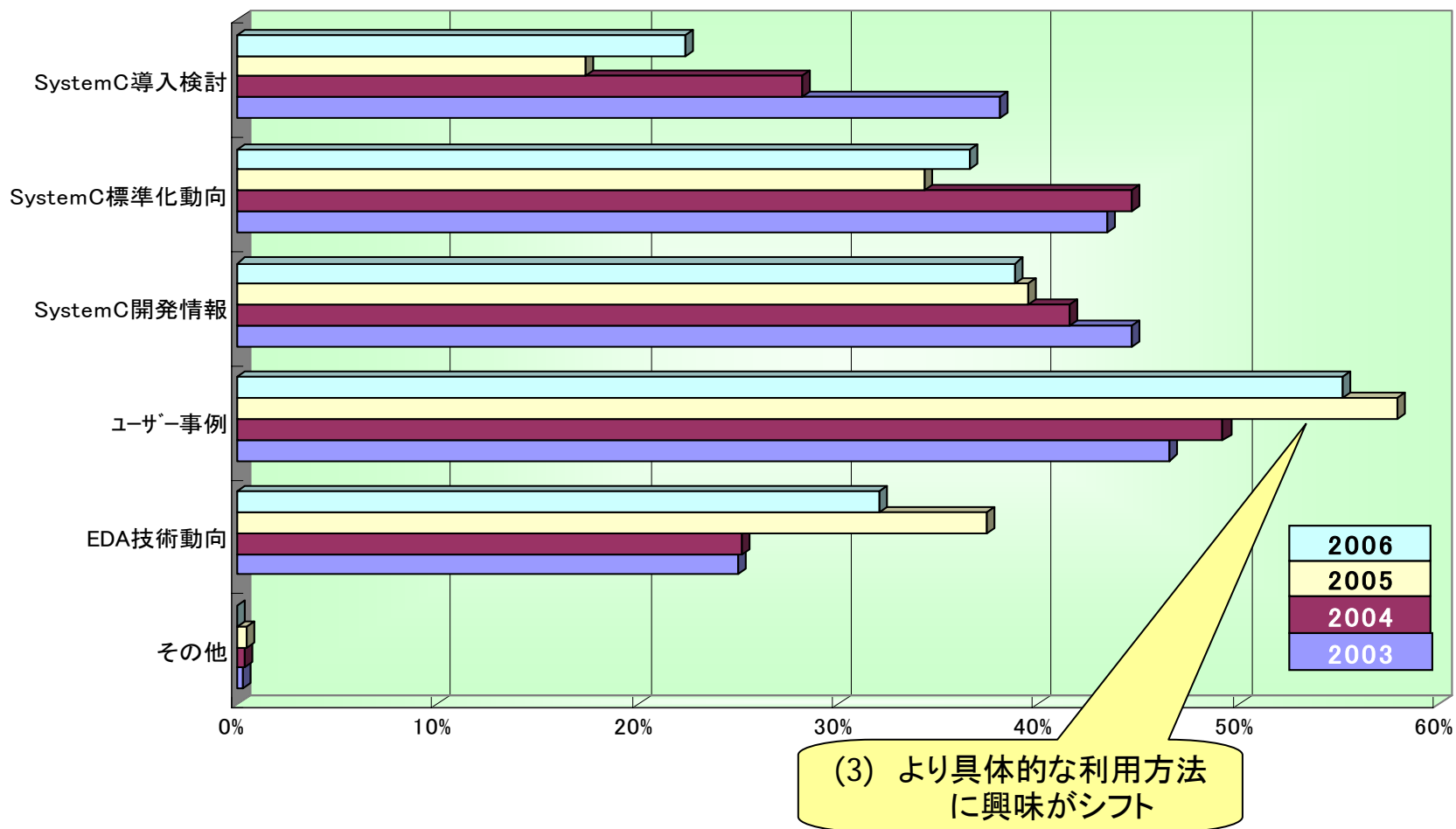


(2) 年度によって構成の大きな変化なし

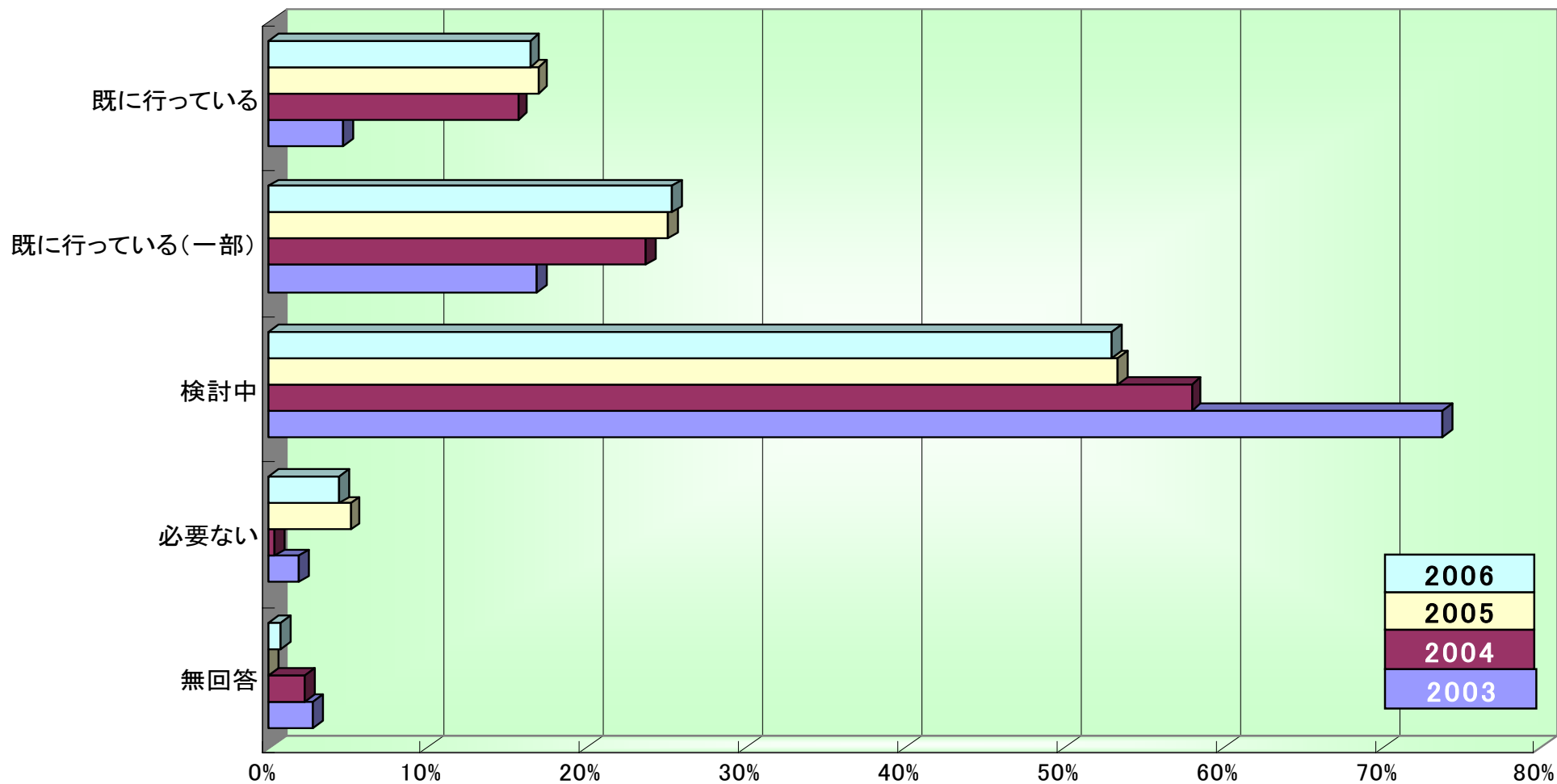
3. 現在主に使用している言語は?



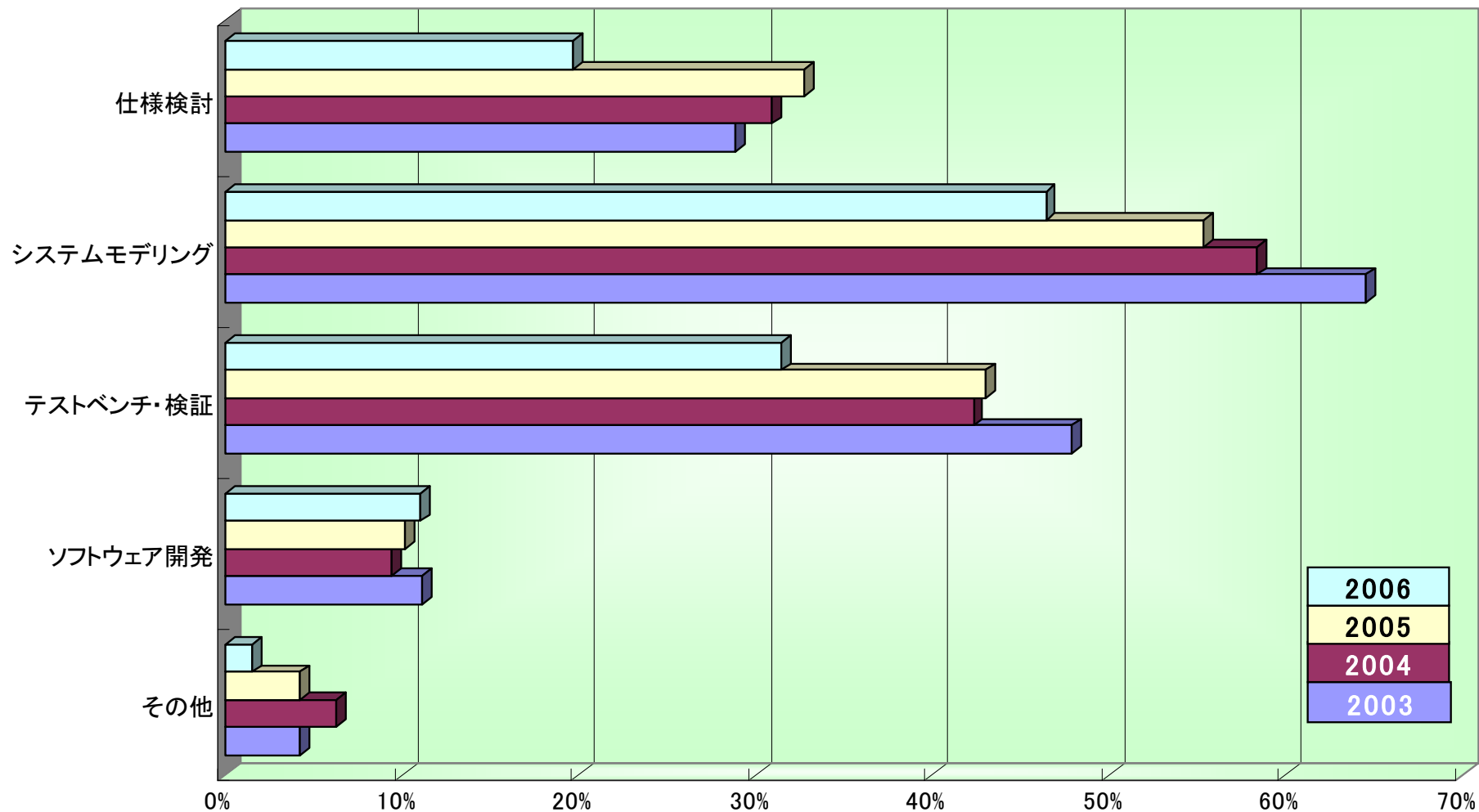
4. SystemCユーザフォーラムに参加された目的は?



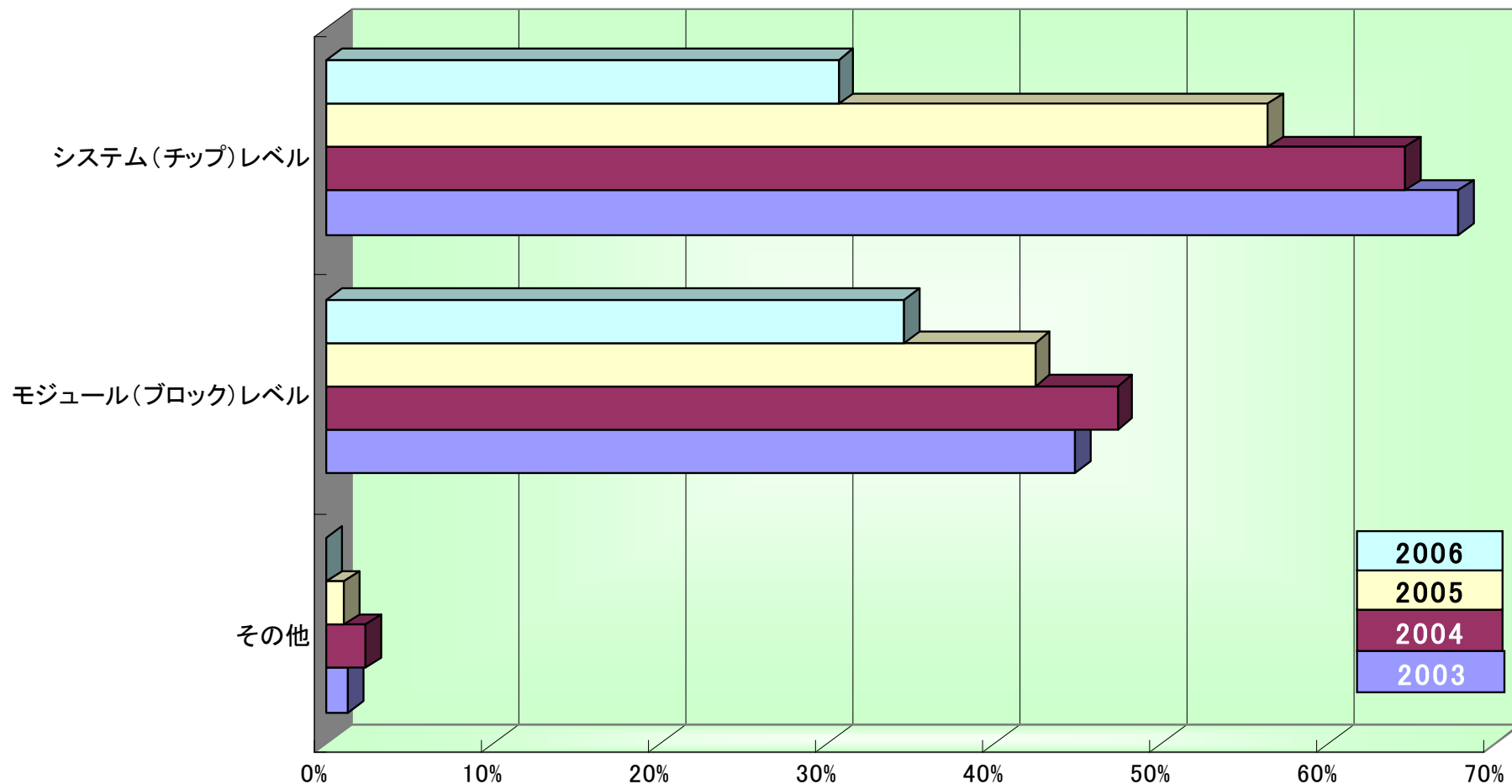
5. SystemCでの設計・検証環境構築について



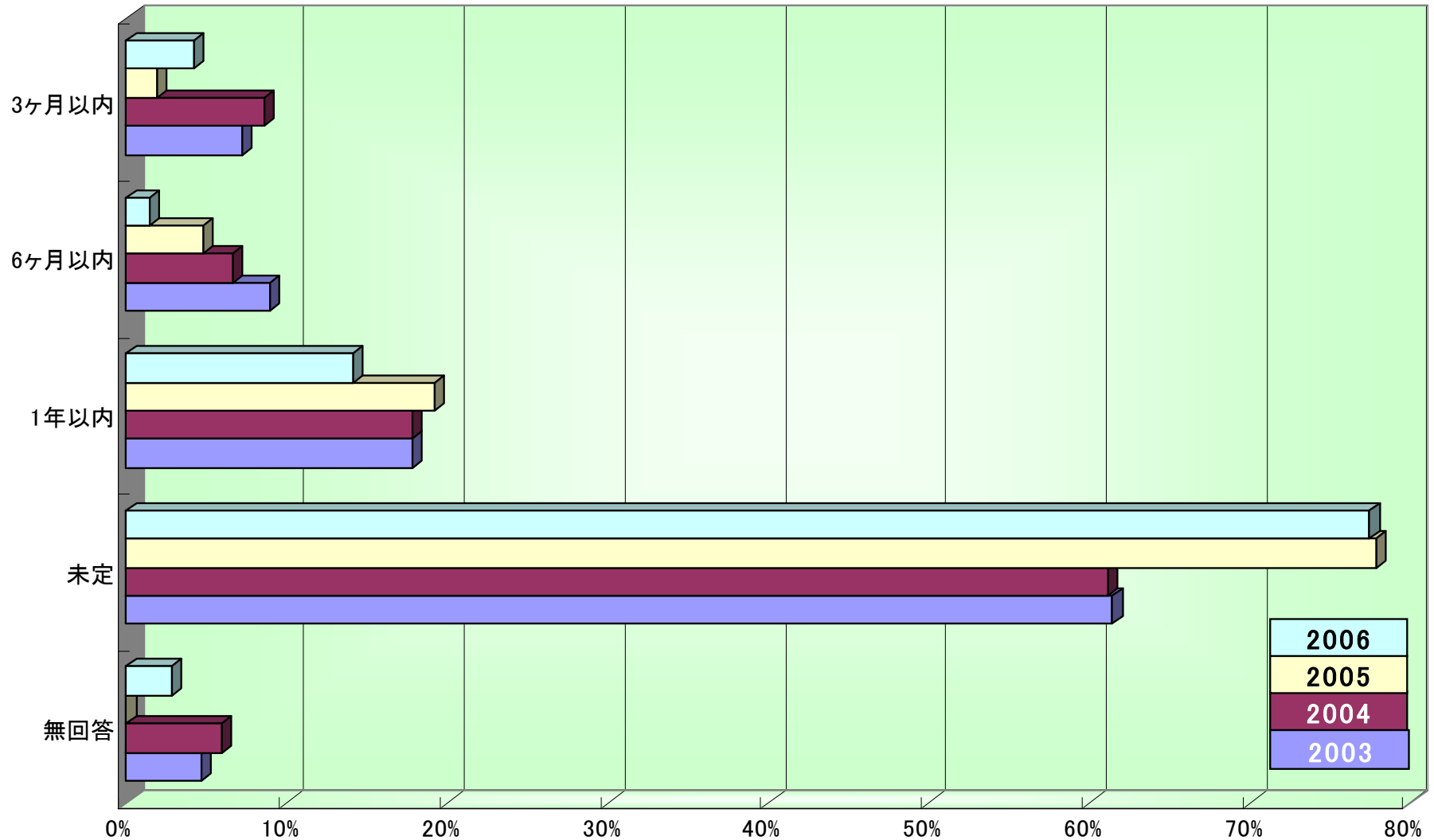
6. 「5」で「既に行っている」または「検討中」と回答された方 a) SystemCの使用目的は?



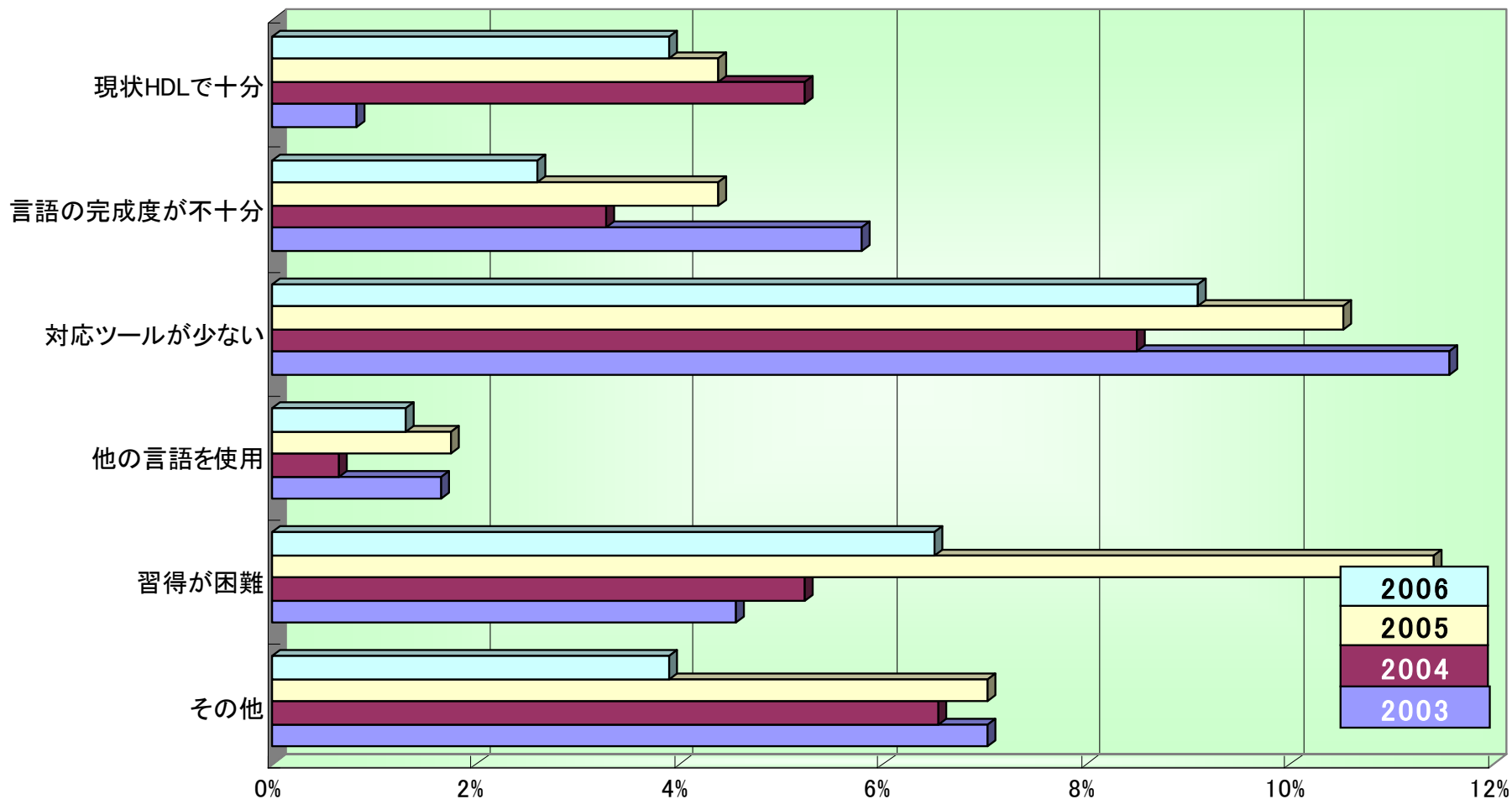
b) SystemCの活用範囲は?



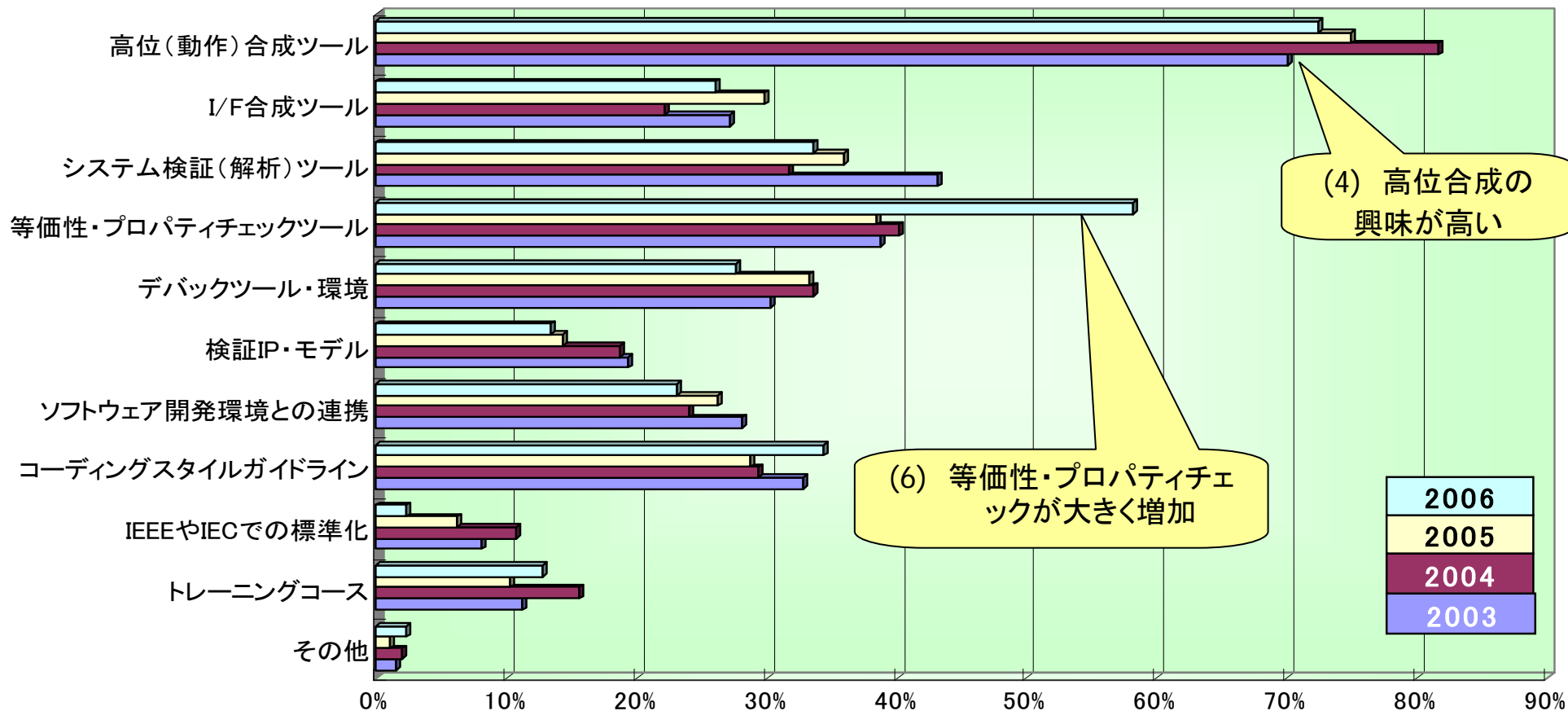
7. 「5」で「検討中」と回答された方へ 導入予定時期は?



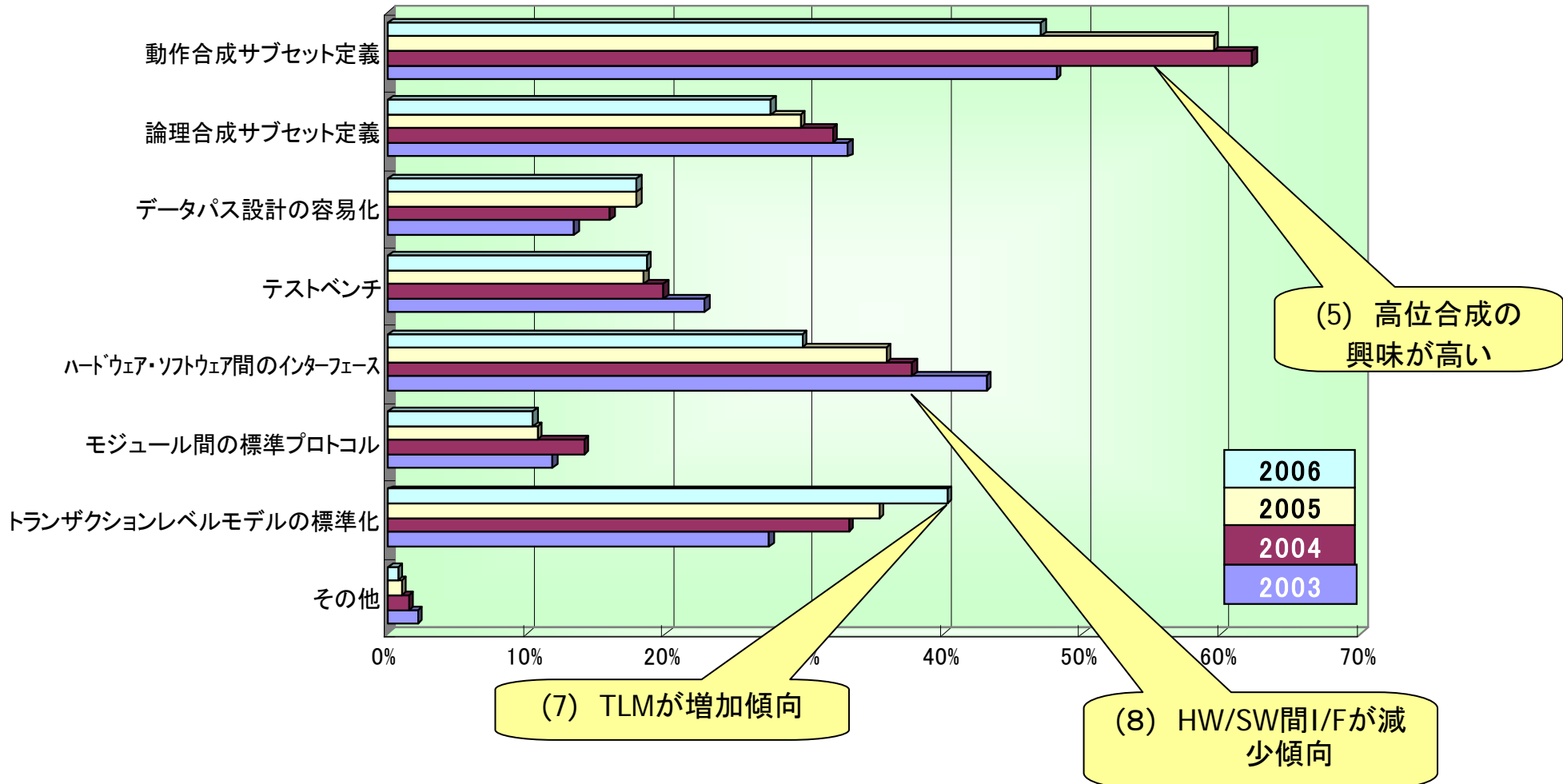
8. 「5」で「必要ない」、「検討中」と回答された方へ導入の障害となっている理由は?

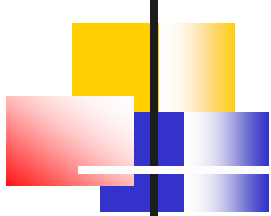


9. SystemCをより活用する為に充実が必要なものは？

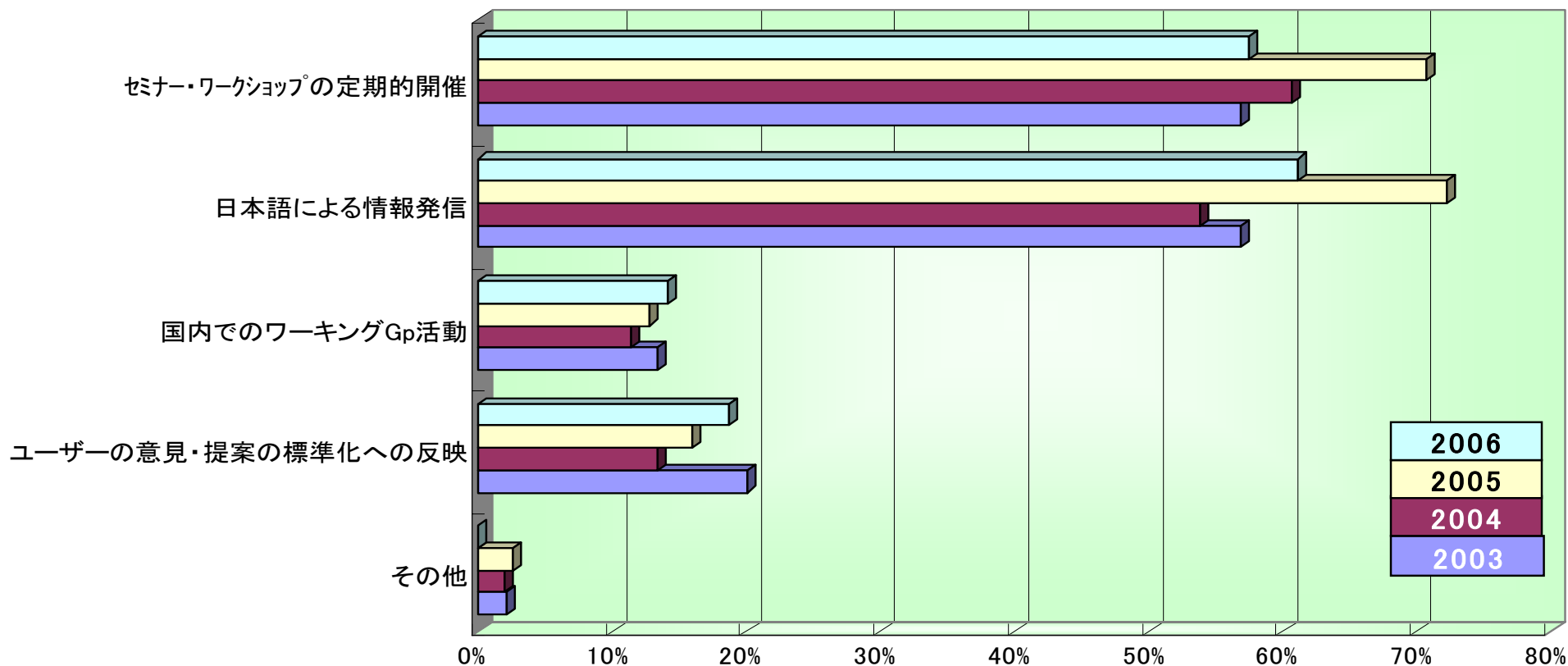


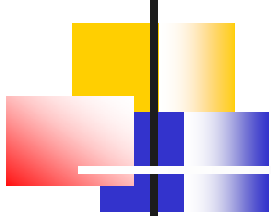
11. 今後SystemCの言語拡張・標準化で期待することは?





12. 今後Open SystemC Initiative (OSCI) や JEITA に期待する活動は？





資料編 2. OSCI-SystemC合成サブセットDraft 1.1.18の概要