

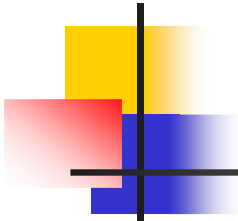
The IEEE P1666 Project : 日本国内におけるSystemC 標準化活動

JEITA EDA技術専門委員会

標準化小委員会

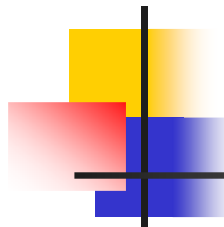
SystemCワーキンググループ

主査：長谷川 隆(富士通マイクロエレクトロニクス)



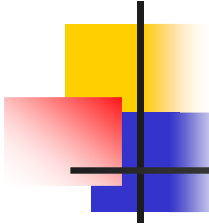
目次

- SystemCワーキンググループとは
- これまでの成果と今後の活動
 - 標準化
 - 動向調査
 - 普及
 - メソドロジ
- 付録



SystemCワーキンググループメンバー

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)
	小島	智	(NECシステムテクノロジー)
	清水	靖介	(OKI)
	逢坂	孝司	(ケイデンス)
	長尾	文昭	(三洋半導体)
	西園寺	修	(シノプシス)
	柿本	勝	(ソニー)
	竹村	和祥	(松下電器)
	牧野	潔	(メンター)
	渡邊	政志	(ルネサステクノロジ)
客員	今井	正治	(大阪大学)
			(2007年度)



SystemCワーキンググループとは

■ 設立

- 2003年10月に、JEITA EDA技術専門委員会 標準化小委員会内に、国内半導体ベンダー及びEDAベンダーからSystemCエキスパートが結集して設置

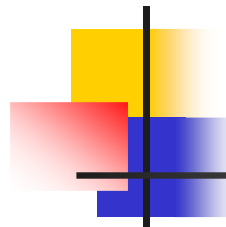
■ 目的

■ SystemC標準化

- 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による設計メソロジの普及

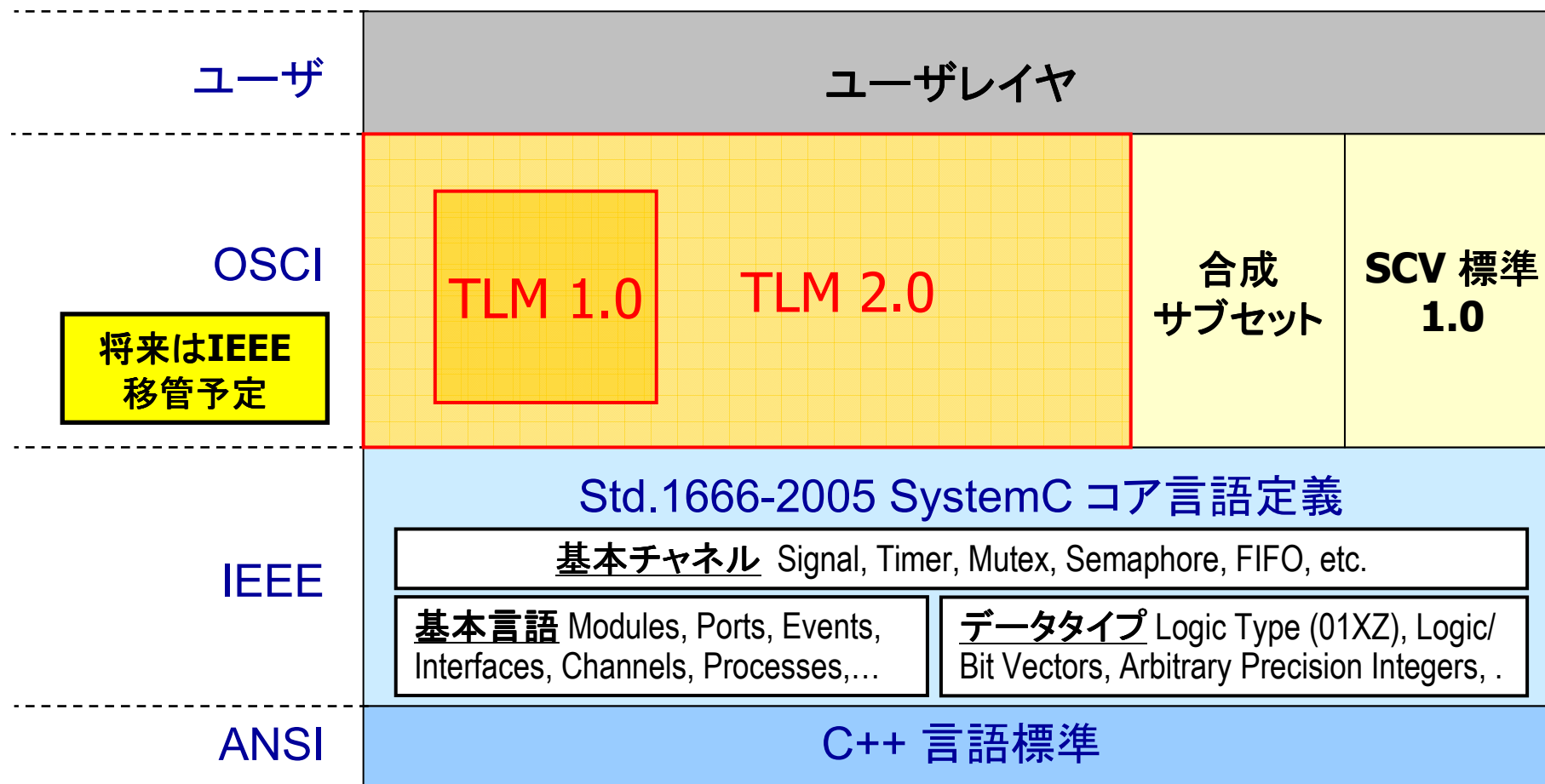
- SystemCに関連した動向調査や技術調査を行い、SystemCユーザフォーラム等で調査結果を積極的に情報発信を行うことで、国内普及を図る。



SystemC とは

- C++言語をベースとした、システムレベル設計言語の代表的な言語である
 - Open SystemC Initiative (OSCI)という標準化組織により、言語仕様(LRM)とリファレンスシミュレータが策定され、無償提供されている
<http://www.systemc.org/>
 - 2005年12月に、SystemC 2.1のLRMがIEEE Std. 1666-2005として標準化された
 - 現在は合成サブセットやTLM(Transaction Level Modeling)の標準化案が検討されている
- C++の文法を保持したまま、クラスライブラリの形で以下のような言語拡張がなされている
 - 並列動作を可能とするシミュレーションエンジン(クロック、イベント、等)
 - 抽象化された通信手段(Channels, Interfaces)
 - ハード実装に必要なデータタイプ (固定小数点、固定長ビット、等)

SystemCの標準化の階層



将来はIEEE
移管予定

OSCI資料(2007)に加筆

SystemC WG活動成果

標準化

- IEEE P1666 WG (2004~2005)
LRM レビューとフィードバック、投票
- TLM 2.0 draft1/2 (2006~2007)
レビューとフィードバック
- 合成サブセット (2006)
レビューとフィードバック

動向調査

- SystemCユーザフォーラム(2003~2008)
出席者アンケート調査
- SystemC 論文調査
(2000~2005)
- TLM動向調査 (2006)
EU vs Japan

SCWG

普及

- SystemC 2.1 新機能解説
と互換性アドバイス
- SystemC-SystemVerilog 共通辞書
(English-Japanese)
- TLM2要求仕様、チュートリアル、合成
サブセットの抄訳

メソドロジ

- “SystemC動作合成
スタイルガイド”構成要件
(2006)
- SystemC推奨設計メソドロジ(合成編)
(2007)

SystemC WG活動成果

標準化

- IEEE P1666 WG (2004~2005)
LRM レビューとフィードバック、投票
- TLM 2.0 draft1/2 (2006~2007)
レビューとフィードバック
- 合成サブセット (2006)
レビューとフィードバック

動向調査

- SystemCユーザフォーラム(2003~2008)
出席者アンケート調査
- SystemC 論文調査
(2000~2005)
- TLM動向調査 (2006)
EU vs Japan

SCWG

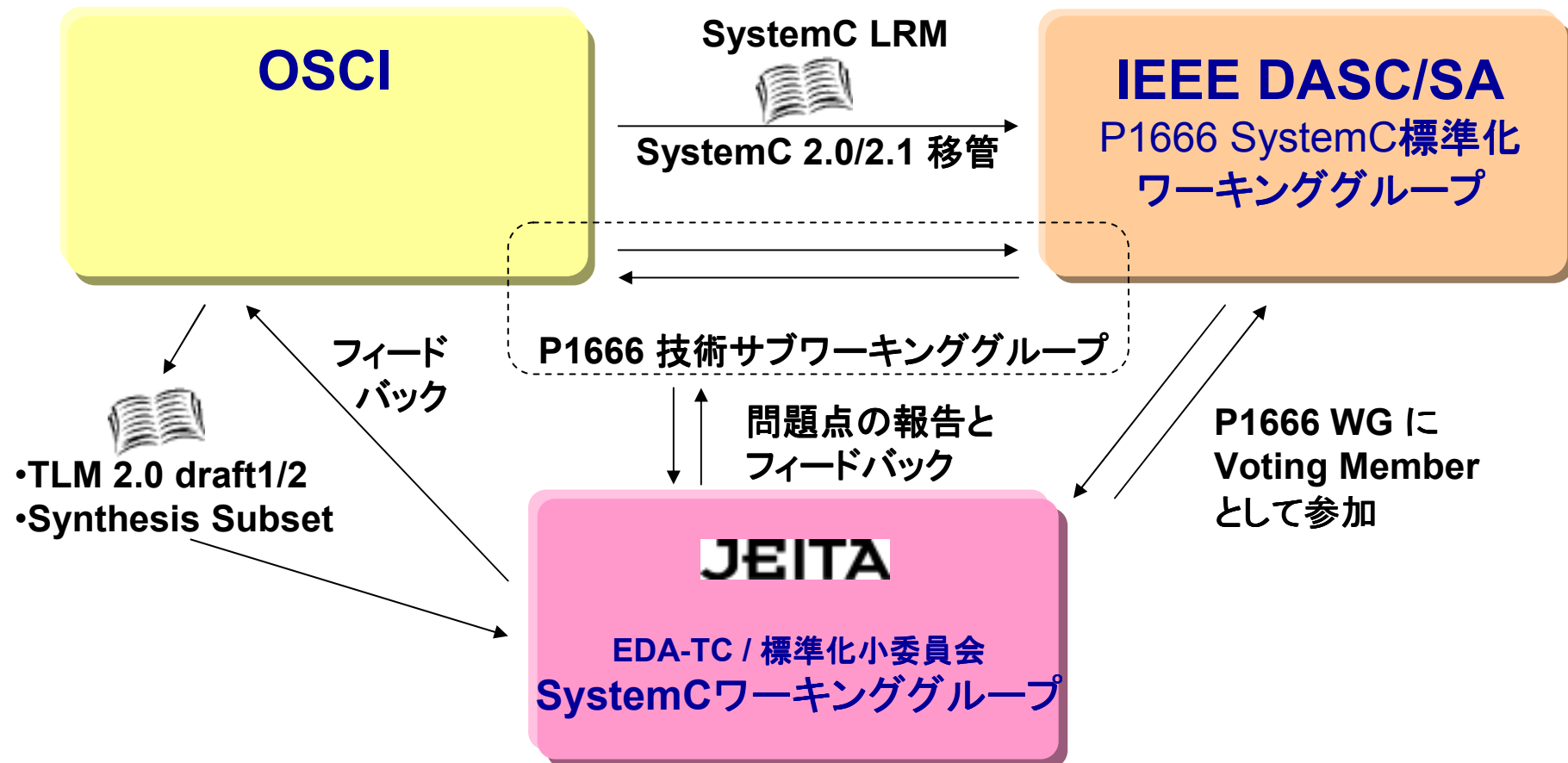
普及

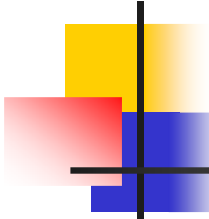
- SystemC 2.1 新機能解説
と互換性アドバイス
- SystemC-SystemVerilog 共通辞書
(English-Japanese)
- TLM2要求仕様、チュートリアル、合成
サブセットの抄訳

メソドロジ

- “SystemC動作合成
スタイルガイド”構成要件
(2006)
- SystemC推奨設計メソドロジ(合成編)
(2007)

SystemC標準化の枠組み





IEEE P1666 メンバー(2005年3月当時)

- Voting Member

- | | |
|------------------|--|
| ■ Cadence | Victor Berman (Chair),
Stuart Swan (Tech.SWG Chair) |
| ■ Calypto | Skip Hovsmith |
| ■ MentorGraphics | Dennis Brophy |
| ■ Synopsys | Oz Levia |
| ■ JEITA | Takashi Hasegawa |
| ■ Jeda | Eugene Zhang |

- Observer

- | | |
|-----------------|----------------|
| ■ Freescale | Rob Slater |
| ■ Tensilica | Grant Martin |
| ■ Summit Design | Vincent Viteau |
| ■ Xilinx | Adam Donlin |
| ■ Pentum | Bodo Parady |

SystemC LRM Issue List (抜粋)

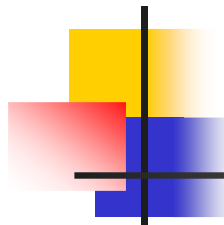
JEITA LRM ISSUE LIST for “Draft Standard SystemC Language Reference Manual” (Released on April 25 2005)					20-May-05
JEITA Issue #	Classification	Summary	LRM 2.1 section	Affiliation	fixed in LRM2.1 ()=Doulos' Comments
1	Language Definition Problem	interface, discrepancy between 3.1.4 and B.2.29	3.1.4	Fujitsu Limited	x
2	Terminology, Grammar, and Typographical Error	definition error of sc_event operator=	5.9.2	NEC Electronics Corporation	x
3	Terminology, Grammar, and Typographical Error	The explanation for to_seconds() includes a typographical error.	5.10.4	NEC Electronics Corporation	x
4	Language Deficiencies and Modeling Problems	sc_event_finder should support multiple parent ports	5.7	NEC Electronics Corporation	(language issue and may be considered in the future)
44	Terminology, Grammar, and Typographical Errors	In the second paragraph, “(or to the base class su-object of class sc_object)”. It should be “(or to the base class sub-object of class sc_object)”	5.6	Toshiba Corporation	x
45	Terminology, Grammar, and Typographical Errors	The class definition of sc_process_handle is different from that of the source code of version oct_12_2004	Annex D.3	Toshiba Corporation	
46	Language Definition Problem	Backward Compatibility with SystemC2.0x for sc_report	8.4.2	Mentor Graphics Japan Co., Ltd.	x
47	Terminology, Grammar, and Typographical Errors	Unnecessary reference	5.12	Matsushita Electric Industrial Co., Ltd.	
48	Examples, Notes, and Appendices	Unnecessary Note description	4.3.2	Fujitsu Limited	
49	Terminology, Grammar, and Typographical Errors	‘whilst’ should be changed to ‘while’.	4.5.5	Fujitsu Limited	
50	Examples, Notes, and Appendices	Explanation of example of unsigned type is not sufficient	7.10.9.12-18	Toshiba Corporation	x
51	Examples, Notes, and Appendices	Wrong third argument of sc_ufixed	7.10.9.13-18	Toshiba Corporation	

IEEE P1666 Final Ballot (抜粋)

P1666/D2.1 April 25th, 2005 Ballot Comments								
Sub-Group	No.	Clause/ Subcla use	Type of Comment (Technical/ Editorial)	COMMENTS (Justification)	Proposed change	Prio rity	Resolution	Approved
JEITA	4	5.7	Technical	sc_event_find er should support multiple parent ports	Current sc_event_finder doesn't support multiple parent ports. This is not good in case of modeling a transaction bus. When there is connection to multiple parent ports in case of tlm port, current sc_event_finder can only find the first parent port even	High	Add the following: "If an event finder member function is called for a multiport object, the implementation shall only find an event for the first channel instance to which the multiport is bound. If the multiport is bound to multiple channel instances, a	Approved (Please add example code)
JEITA	9	6.6.4	Technical	The explanation of update() of "6.6.4 Member	Please add description of the concrete example about operation to the explanation of update() of "6.6.4 Member functions".	High	Add the following: "In other words: suppose the current value of the buffer is V, and member function write is called with argument value V. Function write will store the new value V (in some implementation-defined storage area distinct from the current v	Approved
JEITA	17	6.21.2	Technical	Insufficient explanation of constructor	(The following answer was provided by Andy Goodrich.) First, why the default constructor must be mentioned: there is a copy constructor, and it is private, so unless you define another constructor, C++ will assume there is no way to create an instance of	High	No change. Firstly, the balloter is asking for an explanation of a C++ language issue, which is outside the scope of the SystemC LRM. Secondly, the answer to why the SystemC code is the way it is has been lost in history anyway, but there is no strong re	Approved
JEITA	18	6.22.2	Technical	Insufficient explanation of constructor	(The following answer was provided by Andy Goodrich.) In C++, conversion from double to integer type is well understood: a non-explicit sc_bigint(double) constructor allows an expression with an sc_bigint parameter to be passed a C++ floating point argume	High	No change. See the answer to #17.	Approved
JEITA	47	5.12	Editorial	Unnecessary reference	"(see 5.12)" reference at section 5.12 is strange. This reference should be deleted.	High	Agreed. Delete the reference from this clause to itself.	Approved
JEITA	48	4.3.2	Editorial	Unnecessary Note description (sc_main_mai	Following note doesn't seem necessary. NOTE - Function sc_elab_and_sim was previously named sc_main_main. Reason:	High	Agreed that the use of "previously" is unclear, but the note is still useful. Change the text to read "NOTE - Function sc_elab_and_sim was named sc_main_main in an earlier version of SystemC"	Approved
JEITA	49	4.5.5, Annex	Editorial	whilst -> while	Better to use 'while' rather than 'whilst' since the other part of this document uses 'while'	High	Agreed. Change "whilst" to "while" throughout the entire document	Approved
JEITA	51	7.10.9. 13-18	Editorial	Wrong third argument of sc_ufixed	In the unsigned example, the third template argument is wrong. For example, in the example of SC_RND_ZERO, sc_ufixed<12,8,SC_RND> should be sc_ufixed<12,8,SC_RND_ZERO>	High	Agreed - correct the mistake in the text.	Approved

合成サブセット Issue List (抜粋)

JEITA ISSUE LIST for "SystemC Synthesizable Subset Draft 1.1.18" (Released on December 23, 2004)							Jan. 25, 2007	↓
JEITA Issue #	Clause/ Subclaus	Title	Type of Comment (Technical/Editorial)	COMMENTS (Justification)	Proposed change	Resolution		
1	1.3	Terminology	Technical	Missing description of 'no difference with C++'.	Add explanation of 'no difference with C++'.			不具合・改善要望の分類
2	1.3	Terminology	Technical	Missing description of 'supported but not recognize'.	Add explanation of 'supported but not recognize'. Or, map to the other category.			記述の改善要望
48	A	Syntax summary	Technical	Missing 'ignore' (with underline) description.	Add 'ignore' (with underline) description.			誤記訂正
49	A.2	Lexical conventions	Technical	All the sc_object should accept the 'string' as a name. In other way, 'string' can be converted to the constant.	Remove 'struck-through' on 'string'. (Make 'string' usable in literal.)			技術的改善要望
50		Lexical conventions	Technical	'string' should be treated as an array of 'char' type.	Remove 'struck-through' on 'string'. (Make 'string' usable in literal.)			技術的改善要望
51	A.3	Basic concepts	Editorial	Missing 'Shadow' for 'sc-main-'	Add 'Shadow' for 'sc-main-definition'.			誤記訂正
52	A.4	Expressions	Technical	Destructor should be supported.	TBD			技術的改善要望
53	A.4	Expressions	Technical	Member of static memory should be specified by pointer.	Add : (Make these usable) postfix-expression -> [template] id-expression postfix-expression -> pseudo-destructor-name			技術的改善要望
54	A.5	Statements	Technical	'variable-assignment-statement' and 'procedure-call-statement' are not added in SystemC.	Remove 'Shadow' for 'variable-assignment-statement' and 'procedure-call-statement'.			記述の改善要望
55	A.5	Statements	Technical	Restrictions for 'switch' statement is not reasonable. Should be same as	Remove restriction.			技術的改善要望
56	A.6-1	SystemC Type Specifiers	Technical	'sc_bit' is deprecated in IEEE 1666.	Delete 'sc_bit'.			IEEE 1666対応
57	A.7	Declarators	Technical	Declaration of the pointer for static memory should be supported.	Add : (Make these usable) * [cv-qualifier-seq] [:] [nested-name-specifier] * [cv-qualifier-seq]			技術的改善要望
							誤記訂正	8
							記述の改善要望	24
							技術的改善要望	14
							IEEE 1666対応	11
							計	57

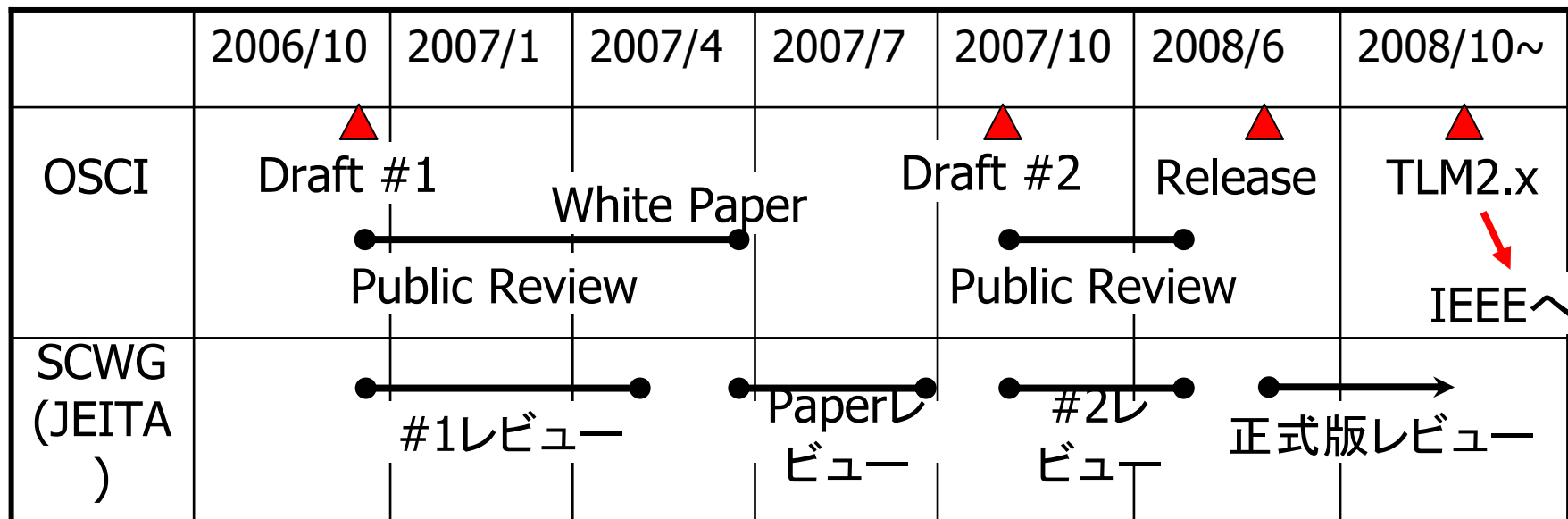


TLM 2.0 draft2 Issue List (2008年1月)

JEITA TLM2.0 draft2 Feedback for “OSCI TLM2 USER MANUAL” (Software version: TLM 2.0 Draft 2, Document version: 1.0.0)			
JEITA Issue #	Classification	Summary	OSCI TLM2 USER MANUAL
1	Language Definition Problem	Difficult to understand how to use TLM_REJECTED. More usage description required.	4.2.2
2	Language Definition Problem	There is no example for Payload Event Queue. Current example is not inadequate because it's using MyPEQ.	8
3	Language Definition Problem	Needs to define the necessary condition for the switchability from LT to AT and/or the other way.	3.3.6
4	Language Definition Problem	Needs to describe clearly whether LT and AT models can be exist together within a system or not.	3.3.6
5	Language Definition Problem	Add the management mechanism for address in the BUS model. It is not clear whether AddressAttribute is absolute value or relative value.	9.9
6	Language Definition Problem	Make clear the difference between Streaming and Burst.	9.14
7	Request	The structure of the section 9.18 and 9.18.1 is not appropriate. The sub-section 9.18.1 should be the separate section within chapter 9 and it should include the explanation for error codes in the section 9.18.	6.6.4
8	Language Definition Problem	Difficult to understand how to use the Analysis port. More usage description required.	6
9	Request	Add code examples in the document, in addition to the separate example package.	-
10	Request	Add more examples for BackwardPath, Debug Interface.	Example

OSCI TLM2.0の標準化スケジュール

第8回NASCUGにおけるOSCIの発表より



- 現在入手可能な資料(OSCIホームページから)
 - TLM要求仕様: Requirements specification for TLM 2.0
 - http://www.systemc.org/projects/tlm/document/TLM2_Requirements.pdf
 - White Paper: Transaction Level Modeling using OSCI TLM 2.0
 - http://www.systemc.org/projects/tlm/document/TLM2_Whitepaper.pdf
 - 用語集: TLM 2.0 Glossary
 - http://www.systemc.org/projects/tlm/document/TLM2_Glossary.pdf

SystemC WG活動成果

標準化

- IEEE P1666 WG (2004~2005)
LRM レビューとフィードバック、投票
- TLM 2.0 draft1/2 (2006~2007)
レビューとフィードバック
- 合成サブセット (2006)
レビューとフィードバック

動向調査

- SystemCユーザフォーラム(2003~2008)
出席者アンケート調査
- SystemC 論文調査
(2000~2005)
- TLM動向調査 (2006)
EU vs Japan

SCWG

普及

- SystemC 2.1 新機能解説
と互換性アドバイス
- SystemC-SystemVerilog 共通辞書
(English-Japanese)
- TLM2要求仕様、チュートリアル、合成
サブセットの抄訳

メソドロジ

- “SystemC動作合成
スタイルガイド”構成要件
(2006)
- SystemC推奨設計メソドロジ(合成編)
(2007)

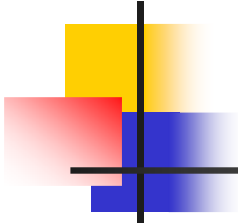


SystemC ユーザ・フォーラム

開催日	時刻	場所(パシフィコ横浜)	主催	協賛	参加費	出席者
2001.2.1	2:00-5:00	Conference Room 414,415	OSCI(US)	OSCI	--	130
2002.1.23	1:20-2:50	Conference Room 414,415	VLD/SLDM	VLD/SLDM	??	> 120
2003.1.31	1:00-5:30	Conference Room 501,502	OSCI(JP)	OSCI	--	420
2004.1.29	1:00-5:00	Conference Room 301,302	OSCI(JP)	OSCI	--	327
2005.1.27	3:20-5:00	Annex Hall F202	JEITA	JEITA/OSCI	--	250
2006.1.27	1:30-3:30	Annex Hall F205,F206	JEITA	JEITA/OSCI	¥ 1600	172
2007.1.26	10:00-12:00	Annex Hall F205,F206	JEITA	JEITA/OSCI	¥ 2000	148
2008.1.25	10:00-12:00	Annex Hall F203,F204	JEITA	JEITA/OSCI	¥ 2100	166

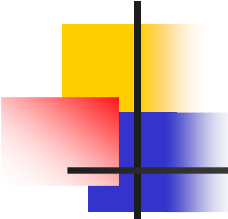
プログラム

	2001	2002	2003	2004	2005	2006	2007	2008
- OSCI Update, including WG Status	x	x	x	x	x	x	x	x
- OSCI Sponsored tutorial	x		x					x
- OSCI Moderated user panel	x							
- User Presentation, Tutorial		x	x	x	x	x	x	x
- Supplier showcase				x	x			
- Survey/Giveaway			x	x	x	x	x	x



SystemCユーザ・フォーラム2008

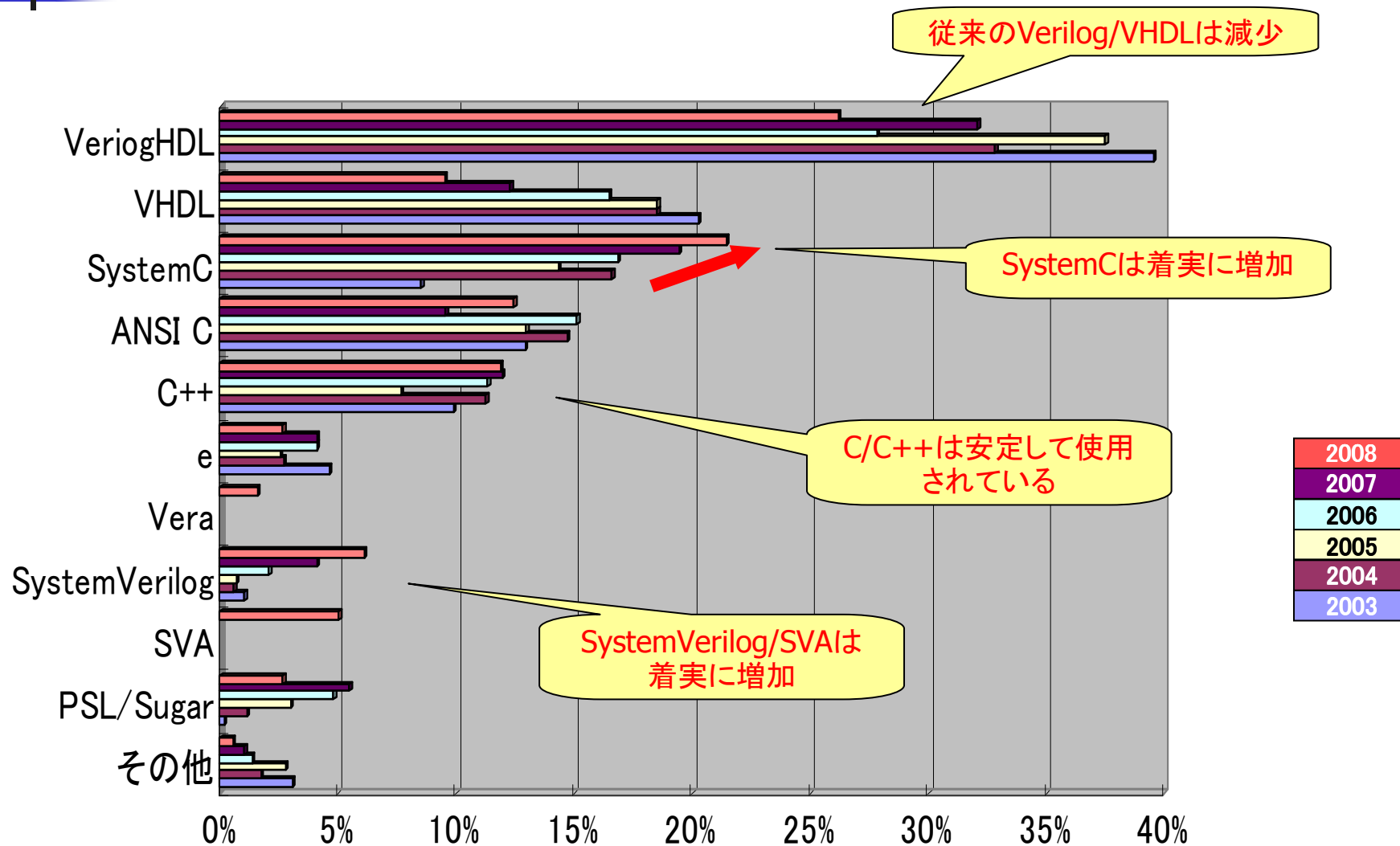
- 主催: JEITA EDA技術専門委員会
- 協賛: OSCI
- 日時: 2008年1月25日 10:00~12:00
- 会場: パシフィコ横浜アネックスホールF203+F204 (定員200名)
- 講演内容:
 - 司会
 - 長谷川 隆氏(SC-WG主査/富士通)
 - 『SystemC Community Update』
 - Patrick Sheridan氏(OSCI/CoWare)
 - 『OSCI TLM2.0 Draft2 tutorial』
 - Throsten Grotker氏(OSCI/Synopsys)
 - 『TLM2.0に関する取り組み/ SystemC推奨設計メソドロジー』
 - 渡邊 政志氏(ルネサス)/長尾 文昭氏(三洋電機)
 - 『TLMを利用したソフトウェア早期開発』
 - 中村 和正氏(富士通)



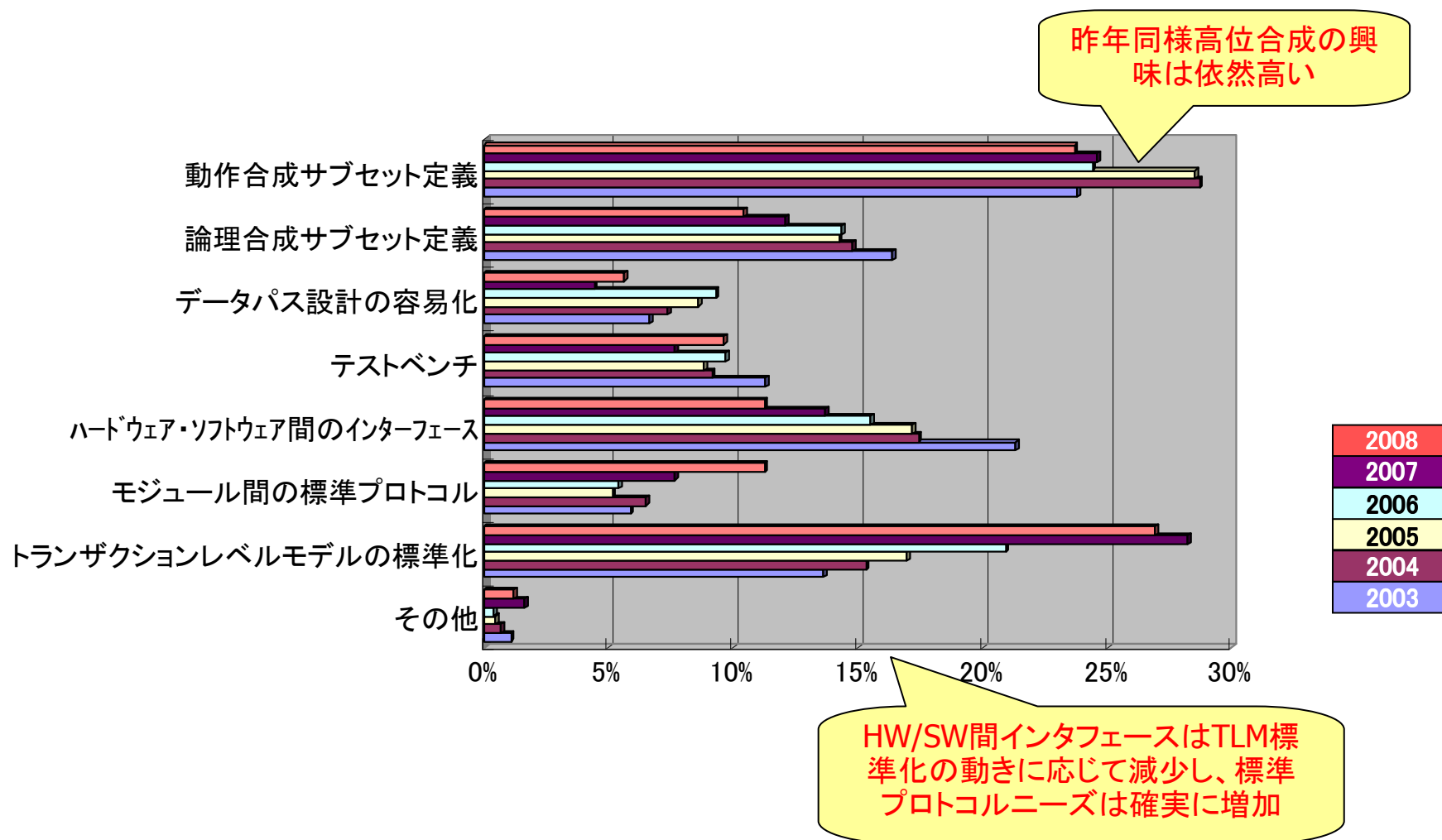
SystemCユーザ・フォーラム2008サマリ

- 登録者数: 172
- 出席者数: 166
- アンケート回答者数: 144
- 出席者の声:
 - 富士通中村氏の発表が素晴らしかった。(同意見計2名)
 - OSCI発表は同時通訳か、日本メンバーに日本語で発表して欲しい。
 - 質問に対するOSCIの回答も日本語訳して欲しい。(同意見計3名)
 - 開発者のチュートリアルが受けれて有意義なフォーラムであった。同フォーラムを次年度以降も続けて欲しい。
 - JEITA渡邊さんの解説で、よりTLMの理解が深まった。
 - 合成も早く標準化して欲しい。

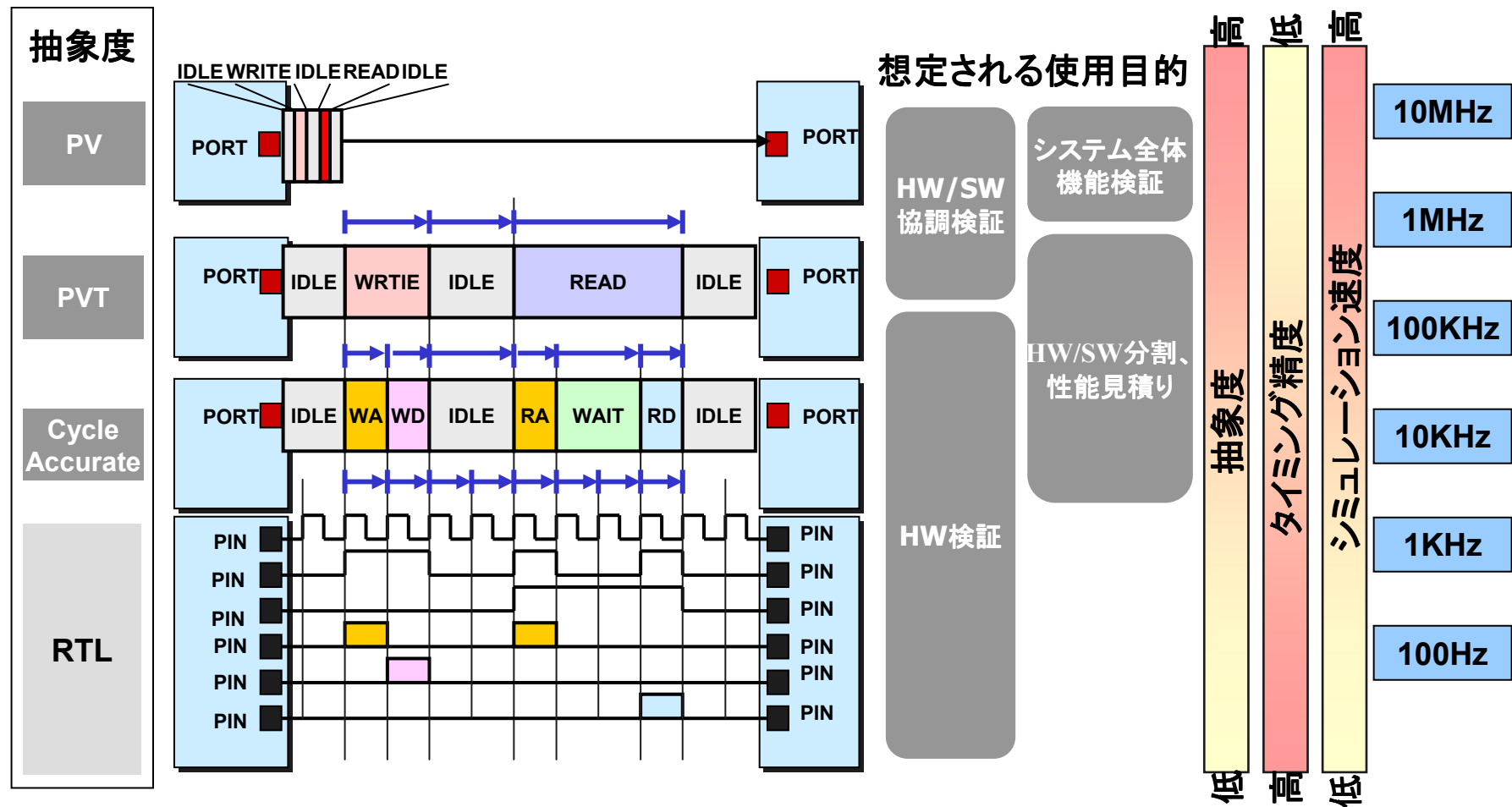
3. 現在主に使用している言語は?



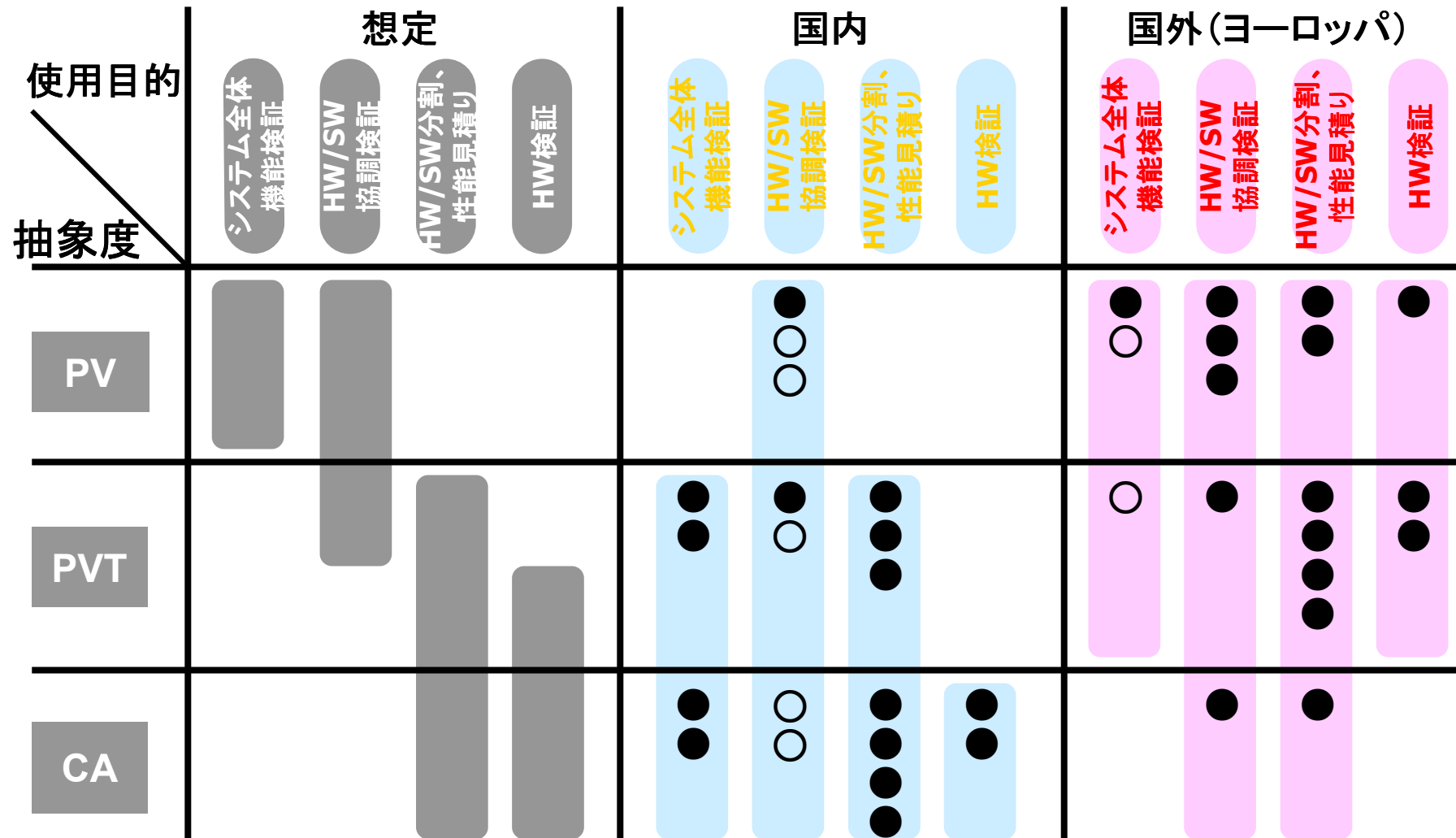
10. 今後SystemCの言語拡張・標準化で期待することは?



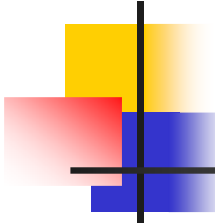
TLMの抽象度定義と想定される使用目的



国内外各社TLM利用状況調査(結果)

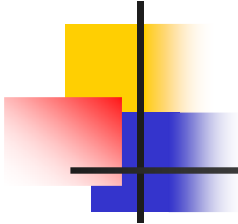


○: C/C++使用、●: SystemC使用
(TLM 1.0の実ユーザーはなし)



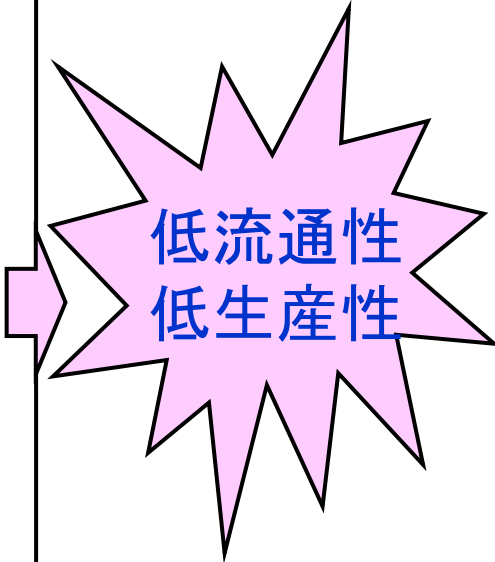
国内外各社TLM利用状況調査(考察)

- 国内、海外とも想定と少しずれているが、HW/SW分割・性能見積り、および、SW設計・検証目的でTLMを利用するケースが多い。
 - OSCI TLM 1.0を利用しているケースは無かった。
- 海外: PVよりにシフト
 - 抽象度の高いTLMを使用しているケースが多い。
 - システム屋さんやソフト屋さんが使っているようだ。(トップダウンの利用。)
- 国内: CAよりにシフト
 - 高抽象度レベルの使用が少ない。ハード設計屋さんが多いため、どうしてもCAモデルを作ってしまうようだ。(ボトムアップの利用。)
 - HW/SW協調検証においてはC/C++でのモデリングの方が多い。



TLM使用に関する現状の課題と展望

- 国内外でのTLMの利用の偏りの原因は？
 - TLMの標準化が未達成
 - 各社独自のTLMを使い、流通性を阻害
 - TLMを使った設計メソロジが未発達
 - HWエンジニアが中々TLMに移行できず生産性が向上しない



低流通性
低生産性

解決策は、TLMの早期標準化と、
TLMベースの設計メソドロジの確立

SystemC WG活動成果

標準化

- IEEE P1666 WG (2004~2005)
LRM レビューとフィードバック、投票
- TLM 2.0 draft1/2 (2006~2007)
レビューとフィードバック
- 合成サブセット (2006)
レビューとフィードバック

SCWG

動向調査

- SystemCユーザフォーラム(2003~2008)
出席者アンケート調査
- SystemC 論文調査
(2000~2005)
- TLM動向調査 (2006)
EU vs Japan

普及

- SystemC 2.1 新機能解説
と互換性アドバイス
- SystemC-SystemVerilog 共通辞書
(English-Japanese)
- TLM2要求仕様、チュートリアル、合成
サブセットの抄訳

メソドロジ

- “SystemC動作合成
スタイルガイド”構成要件
(2006)
- SystemC推奨設計メソドロジ(合成編)
(2007)

SystemC 2.1の新機能(抜粋)

対象LRM: **IEEE P1666/D2.1.1, October 17, 2005**

- **新しいイベントクラス**: 同一サイクル内のイベントも全て実行、ポートを介したモデル間伝達
- **sc_report**: 大幅に見直されたユーザ定義のレポート出力I/F
- エラボレーション終了前、Sim開始時、Sim終了時に**コールされる仮想関数**を追加
- **コマンドライン引数**がどこからでも参照可能に
- sc_mainに加え、**mainを最上位**にすることも可能に
- コンパイル時に参照したSystemCバージョンが異なった場合の**リンク時エラー機能**
- sc_uint/sc_int/sc_biguint/sc_bigintを**キャストなしで連結**することが可能
- SystemC 2.0.1では β 仕様であった**fork/joinが正式サポート**
- **sc_stop()**を**コール**した時にデルタ遅延後にsc_mainに戻るか即時で戻るかを選択可能
- チャンネルをモジュール内部で定義するための**sc_export**を追加

(2005年12月時点での情報)

SystemC2.0.1の互換性のためにSystemC2.1リファレンスシミュレータ(Ver2.1 Beta Oct 12 2004)には実装されているが、LRM(IEEE P1666/D2.1.1, October 17, 2005)では記載されていない機能(注意が必要なものを抜粋)。 将来サポートされない可能性が高いので、今後は使用しない方がよい。

概要	注意度	将来なくなる記述	代替/推奨記述
sc_cylceとsc_initializeを止めてsc_startを使う	◎	sc_cycle, sc initailize	sc_start
シミュレーション状況の情報をsim_contextを直接アクセスせずに、新たに定義された下記アクセス関数を利用する。 <アクセス関数> sc_delta_count sc_is_running sc_get_top_level_objects sc_find_object <メンバー関数> get_child_objects get_parent object	○	sc_simcontext	<function> sc_delta_count sc_is_running sc_get_top_level_objects sc_find_object <member function> get_child_objects get_parent_object
event.notify_delayed()の代わりに、event.nitify(SC_ZERO_TIME)を利用	○	notify_delayed	notify(SC_ZERO_TIME)
operator, and operator<< of class sc_module for positional port binding (Use operator() instead.)	◎	ポート接続の“,”と“<<”	ポート接続の“()”
operator() of class sc_module for positional port binding when called more than once per module instance (Use named port binding instead.)	◎	ポート接続のポジション指定	ポート接続の名前指定
sc_sensitive(clk)をやめて、sc_sensitive << clk を推奨	◎	sensitive(event)	sensitive << event
sc_sensitive(clk)をやめて、sc_sensitive << clk を推奨	◎	sensivei_pos(clk)/neg(clk)	sensitive<< clk.posedge()
default time unit(=1ns)自体の設定変更ができなくなる。 time unitの変更は可能。	○	sc_simulation_time sc_set_default_time_unit sc_get_default_time_unit sc_start(double) sc_clock(const char*, double, double, double, bool)	NONE
sc_signalのメンバー関数get_new_value()を使わない。	○	sc_signalのメンバー関数 get_new_value()	NONE
SYSTEMC_VERSION定数を使わずに、sc_version()関数を使う。	○	定数 SYSTEMC_VERSION	sc_version関数
出力できる波形ダンプファイル形式は、VCDだけ	○	wifフォーマット isdbフォーマット	vcdフォーマットonly
sc_bitを使うよりC++のboolを使ったほうが高速？	◎	sc bit	bool
watching(reset)の代わりに、reset signal is を使う	○	watching in GTHREAD	reset signal is関数

SystemC-SystemVerilog 共通辞書(抜粋)

SystemC/SystemVerilog 対訳表	
keyword or phrase	訳語
!=? (注SV)	ワイルドカード非等価演算子
##0 (注 SV)	シーケンス・オーバーラップ接続
.* port connection	.*ポート接続
.name port connection	.name ポート接続
?: (注SV)	条件演算子
[=n] (注 SV)	シーケンス非連続的繰り返し
[->n] (注 SV)	シーケンスgoto繰り返し
=> (注 SV)	非オーバーラップ含意演算子
-> (注 SV)	オーバーラップ含意演算子
==? (注SV)	ワイルドカード等価演算子
abstract base class	抽象基底クラス
abstract class	抽象クラス
active region	アクティブ領域
aggregate expression	集合体表現
aggregate type	集合体型
alias	エイリアス
antecedent	前提部
application	アプリケーション
arc coverage	アーク・カバレッジ
arc transition	アーク遷移
architecture exploration	アーキテクチャ探索
argument	引数
argument mode	引数モード
argument passing	引数渡し
argument passing by name	引数の名前渡し
array literals	配列リテラル

合成サブセットの抄訳（冒頭部分）

4.2.4.1.2 OSCI-SystemC 合成サブセット Draft 1.1.18 の概要

本ドキュメントは OSCI の Synthesis WG が作成した「SystemC Synthesizable Subset Draft 1.1.18」について、概

要を説明したものである。内容については特に重

いては、原文を参照のこと。本文中の章番号は、

また、文中の「下線+太字」で書かれている部

Synthesis WG にフィードバックをおこなった。

OSCI へ提出した SystemC 合成サブセット Draft

..

1. 概要

1.2 目的

- 合成可能な SystemC 構文（サブセット）
- 動作合成&論理合成を対象としている
- コーディングガイドラインではない。
- Basic Requirements of SystemC Subset for Synthesis (V2.0) になっている。

1.3 用語

- 本ドキュメント内の SystemC 構文の分類
 - ✓ Supported: コンストラクト(構文)をそのままサポート
 - ✓ Extended: C++ 構文を拡張して処理
 - ✓ Ignored: コンストラクト(構文)を無視
 - ✓ Not Supported: コンストラクト(構文)を無視される場合もある。(実装依存)

> (1) no difference with C++ の説明が必要

> (2) Supported but not recognize の表

3. モジュール

3.1 モジュール定義

- モジュール定義は以下のいずれかである。
 - ✓ SC_MODULE マクロの使用。
 - ✓ sc_module クラスからの直接派生。
 - ✓ sc_module クラスからの間接派生。
 - ✓ template 引数を持つモジュール定義もサポート。

3.1.2 モジュール宣言

- 共有メンバ変数（信号やポートではない）は合成不可。
- モジュール内で宣言された変数は同一モジュール内のただ 1 つのプロセスからのみアクセス可能。
- コンストラクタは SC_CTOR と普通の関数形式のどちらもサポートする。
- SC_CTOR を使用する場合、SC_HAS_PROCESS を使用してはいけない。

3.1.2.1 ポートと信号

- サポートしているものは以下の通り。
 - ✓ sc_signal, sc_in, sc_out, sc_inout, sc_in_clk, sc_out_clk, sc_inout_clk。
 - ✓ sc_signal_resolved, sc_in_resolved, sc_out_resolved, sc_inout_resolved。
 - ✓ sc_signal_rv, sc_in_rv, sc_out_rv, sc_inout_rv。

3.1.2.2 モジュール・コンストラクタ

- 例に挙げるような、以下のいずれかである。
 - ✓ SC_CTOR マクロを使って記述した場合。
 - ✓ SC_CTOR マクロを使わずに記述した場合。
 - この場合、少なくともモジュール名は引数として渡す必要あり。

例 1.

```
SC_MODULE( FullAdder ) {  
    sc_in< bool > X, Y, Cin;  
    sc_out< bool > Cout, Sum;  
    SC_CTOR( FullAdder ) {  
    }  
};
```

SystemC WG活動成果

標準化

- IEEE P1666 WG (2004~2005)
LRM レビューとフィードバック、投票
- TLM 2.0 draft1/2 (2006~2007)
レビューとフィードバック
- 合成サブセット (2006)
レビューとフィードバック

SCWG

動向調査

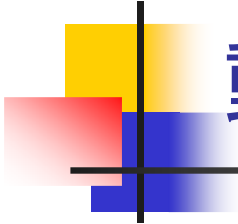
- SystemCユーザフォーラム(2003~2008)
出席者アンケート調査
- SystemC 論文調査
(2000~2005)
- TLM動向調査 (2006)
EU vs Japan

普及

- SystemC 2.1 新機能解説
と互換性アドバイス
- SystemC-SystemVerilog 共通辞書
(English-Japanese)
- TLM2要求仕様、チュートリアル、合成
サブセットの抄訳

メソドロジ

- “SystemC動作合成
スタイルガイド”構成要件
(2006)
- SystemC推奨設計メソドロジ(合成編)
(2007)



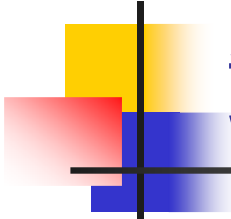
動作合成スタイルガイドの必要性

■ 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設計データの再利用性を高める

■ 動作合成スタイルガイド構成要件の目的

- 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
- ユーザ視点でスタイルガイドに記載する項目を規定
- ベンダ固有のノウハウに依存しない設計データの蓄積
- ツール開発側とユーザー側に基本的な活用スタイルを提案



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

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

第3章 動作合成のための記述スタイル

第4章 検証環境

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

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

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構成要件(抜粋)

<導入部>..

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 動作合成を考慮した SoC 設計..

第6章 RTL との対応..

第7章 上級テクニック..

第8章 SystemC の合成サブセット..

付録別冊 ツールの特徴(ツールに独自な特

3.2. 各章の詳細説明

ここではスタイルガイドの各項目で記載す

3.2.1. 「第1章 概論」

この章では、高位設計の進展や動作合成が読者や設計マネージャが理解できることを目的と捉えて、設計工程に動作合成を取り込むこととする。

以下の項目が含まれて記載されている事とする。

(1) 高位設計が必要となってきた背景..

半導体技術の微細化の進展に伴う SoC 設計ム全体を見渡した設計が必要になってきている役割が大きくなっているなかで、SoC を手戻について説明する。必要に応じて、ツールの紹

(2) C/C++及びその拡張言語である SystemC

高位設計の活用には、アルゴリズムからの記述の再設計(改版、作り直し)など、いろいろな形態などで記述されたアルゴリズムからハードウェアまでの設計の手順を示す。各設計段階 TLM(Transaction Level Model)の説明や、各記載されていることが望ましい。アルゴリズム

3.2.3. 「第3章 動作合成のための記述スタイル」

この章では、初めて動作合成を行う設計者が一通りの動作合成作業が作業できるようなレベルを目標として、合成対象回路の SystemC のモデリング手法と最適化の為の合成制約について解説を行う。全体的には代表的な回路例や記述例を中心に解説を行い、様々なバリエーションや回路のチューニングテクニック等は上級編にて記載する事とする。記載内容としては、以下の3点が含まれていることとする。

(1) 基本的な動作合成向き SystemC の記述スタイル

動作合成対象回路のモジュール記述の基本形を解説する。含まれる内容としては、以下の様なものが挙げられる。

- モジュール記述/プロセス記述/ポート構成
- SC_THREAD プロセスの定義方法とリセット動作の実装方法
- よく使うデータタイプの解説と定数の記載方法

これらはテンプレートとして用いられる事を想定した、実際の SystemC 記述例と共に解説することが望ましい。

(2) 動作合成を実施する際の合成制約の設定方法

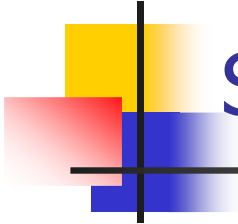
動作合成対象に対する合成制約について、最低限必要なもの/設定すべきものに対して、その制約が合成実行時に回路に与える意味と、その設定方法について解説する。なお合成制約はツール依存性が強い部分ではあるが、なるべく一般化して解説することが望ましい。

(3) 動作合成対象回路の回路構成

動作合成で回路を生成する場合に、特に考慮すべき回路構成について、その動作と実装記述例の解説を行う。含まれるべき回路構成としては以下のようなものが挙げられる。

- パイプライン構造のデータパス回路の構成の実装方法
- 合成対象回路と他のモジュールとのインタフェース回路の実装方法(ピン実装やクラス化)やハンドシェイクの例
- 動作合成回路が使用するメモリ(RAM)の実装方法や取り扱い方

なお、これらの回路構成について様々な場合を記述するのではなく、代表的でかつ比較的シンプル



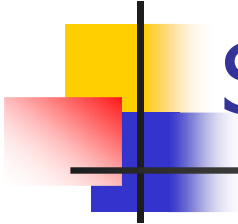
SystemC推奨設計メソドロジの目的と計画

■ 目的

- SystemCの特徴を生かした設計メソドロジ(スタイル)の提案を行い、設計メソドロジの共通認識を広める。
- 設計メソドロジの議論の土台を作る。
- 設計メソドロジ案を公開し、それに対応してもらうことで各種ツールの親和性を高めたい。

■ 計画

- 2008年3月 合成編完成
- 2008年夏 合成編公開
- 2008年度 設計メソドロジ全般を検討予定



SystemC推奨設計メソッドロジの内容

- 主にデジタル信号処理系を対象とする。
- アルゴリズムをHWとSWに分割し、コンパイル可能なレベルに実装するまでの手法を検討する。

全般

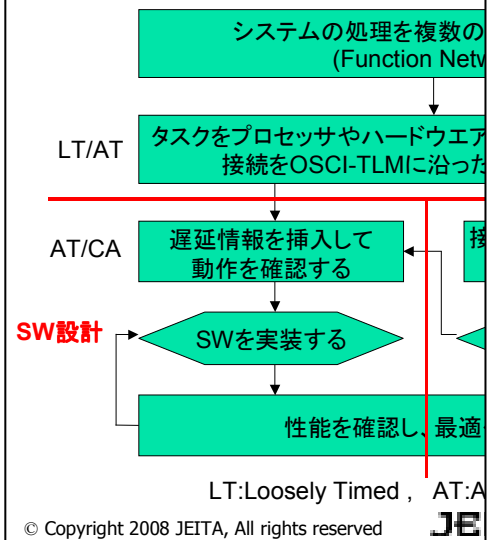
1. 概要
2. システム設計
3. SW設計
4. HW設計 (合成編)
5. 検証
6. モデリング (高速化)
7. その他

合成編

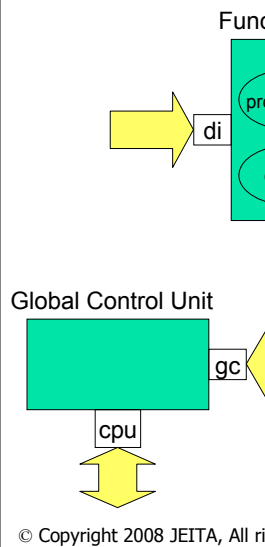
1. FNDMと合成
2. インタフェース
3. マルチ入力インタフェース
4. メモリアクセス
5. 演算系
6. アルゴリズムとアーキテクチャ

SystemC推奨設計メソッドロジ合成編(抜粋)

システムLSIの開発ステップ



システム



インタフェース・ライブラリ

- モジュラーインタフェースで用意される。
- `mi::mi_get_if<TD>`クラスか`mi::mi_put_if<TD>`クラスを継承している。
- ハンドシェイクのみでバッファされないデータ転送
`fn_rdyvld_ch<TD>` チャンネル
`fn_rdyvld_ch<TD>::in` データ入力ポート
`fn_rduvld_ch<TD>::out` データ出力ポート
- レジスタで構成されたFIFOを持つデータ転送
`fn_fifo_ch<TD,FL>` チャンネル
`fn_fifo_ch<TD,FL>::in` データ入力ポート
`fn_fifo_ch<TD,FL>::out` データ出力ポート
- SRAMで構成されたFIFOを持つデータ転送
`fn_fifo_sram_ch<TMW>` チャンネル
`fn_fifo_sram_ch<TMW>::in` データ入力ポート
`fn_fifo_sram_ch<TMW>::out` データ出力ポート
- FIFOのチャンネルには`clk`(立ち上がり動作)と`rst`(負論理)のポートがあり、クロックとリセットの供給が必要である。

© Copyright 2008 JEITA, All rights reserved

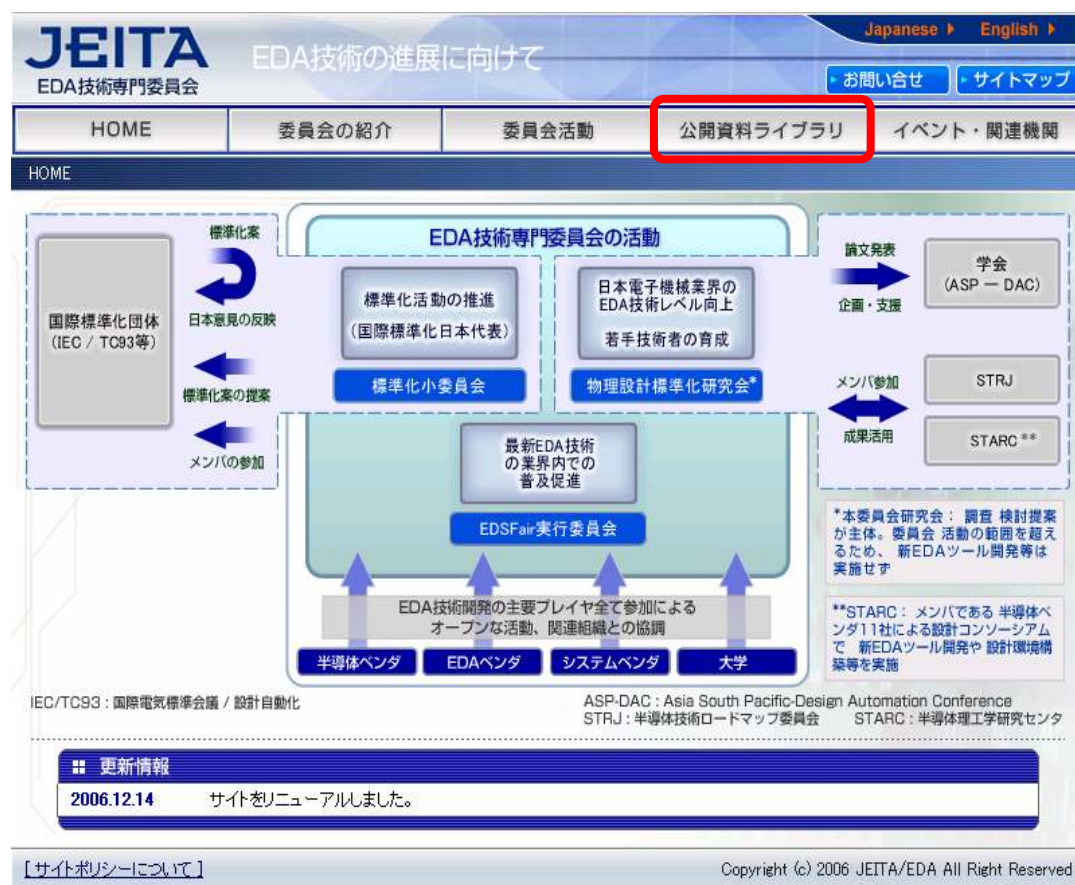
JEITA

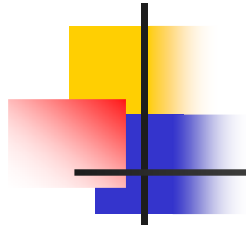
76

活動資料の公開について

EDA技術専門委員会 ホームページ

<http://eda.ics.es.osaka-u.ac.jp/jeita/eda/lib/index.html>

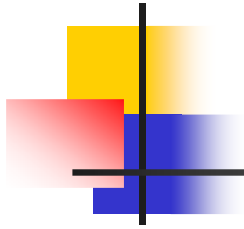




まとめ

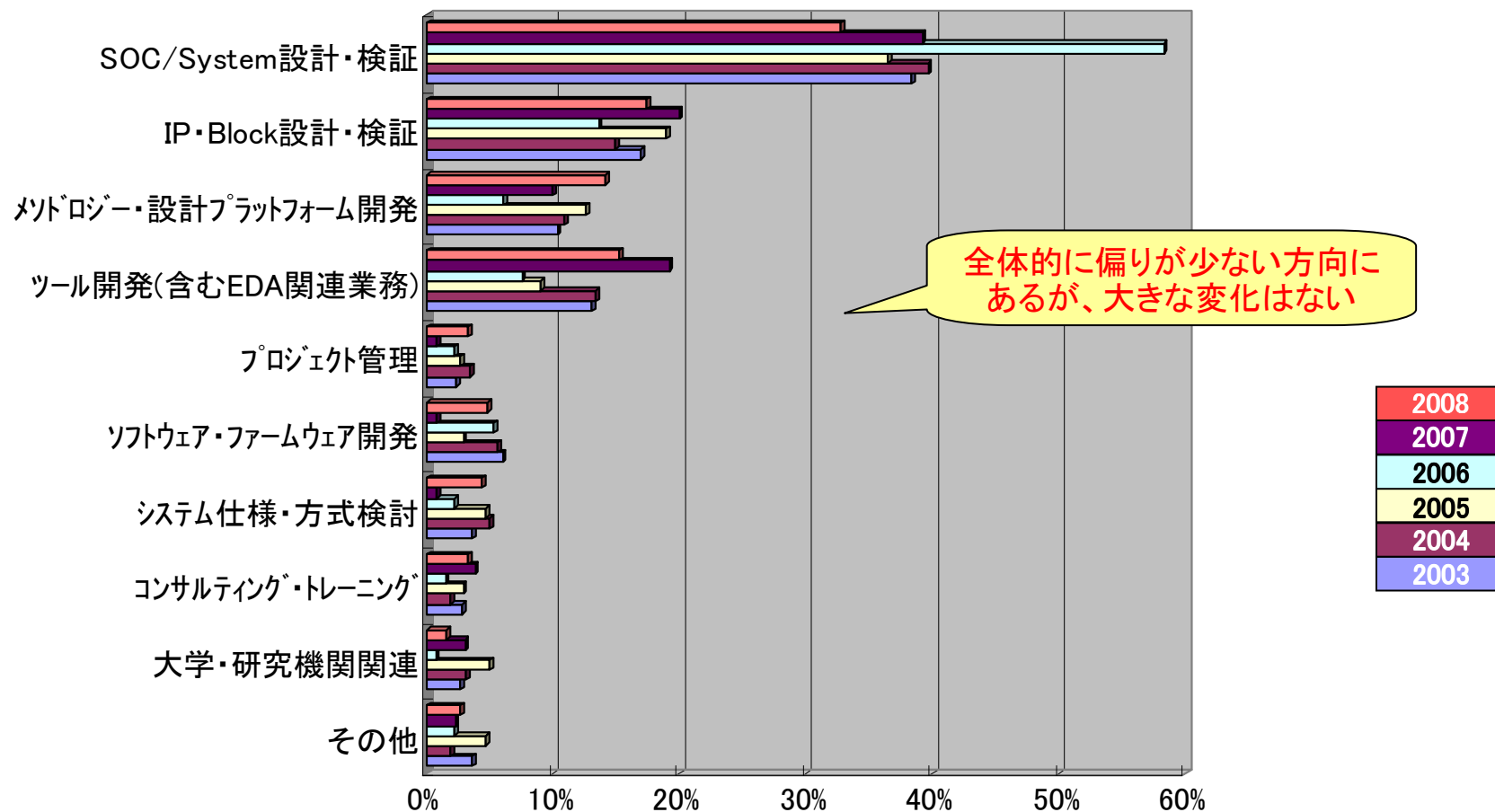
- SoC設計のため、SystemC は既になくてはならない言語になっています
- **OSCI TLM2.0** (draft2以降)による相互利用性の向上が期待されます。
 - JEITAは今後も標準化活動に貢献します！
- **SystemC推奨設計メソッド**により上流設計の普及促進を進めていきます。
 - 標準化活動に加えて、普及活動も強化します！

製品開発に役立ててください！！

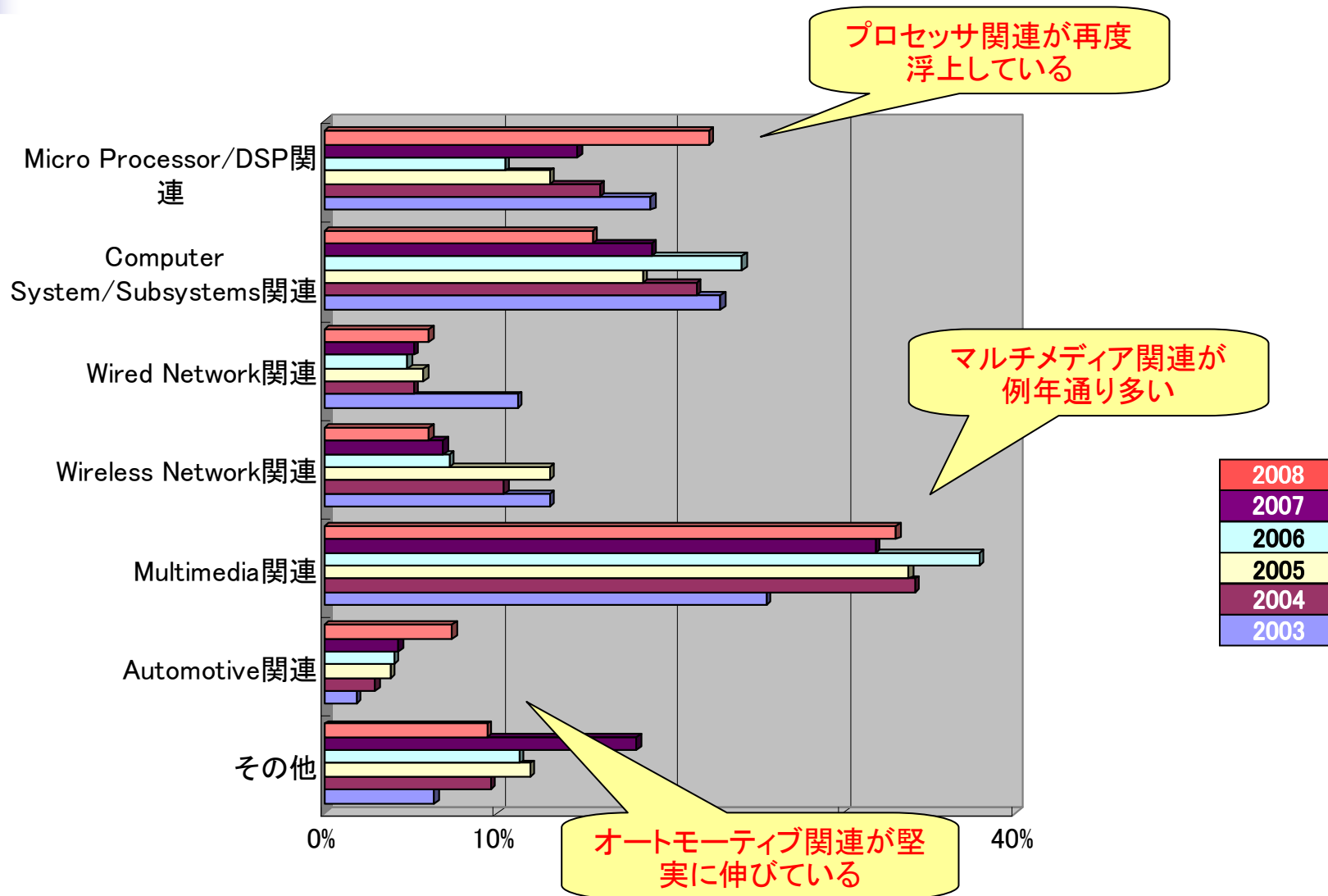


付録:
SystemCユーザ・フォーラム
アンケート調査結果
(2003-2008)

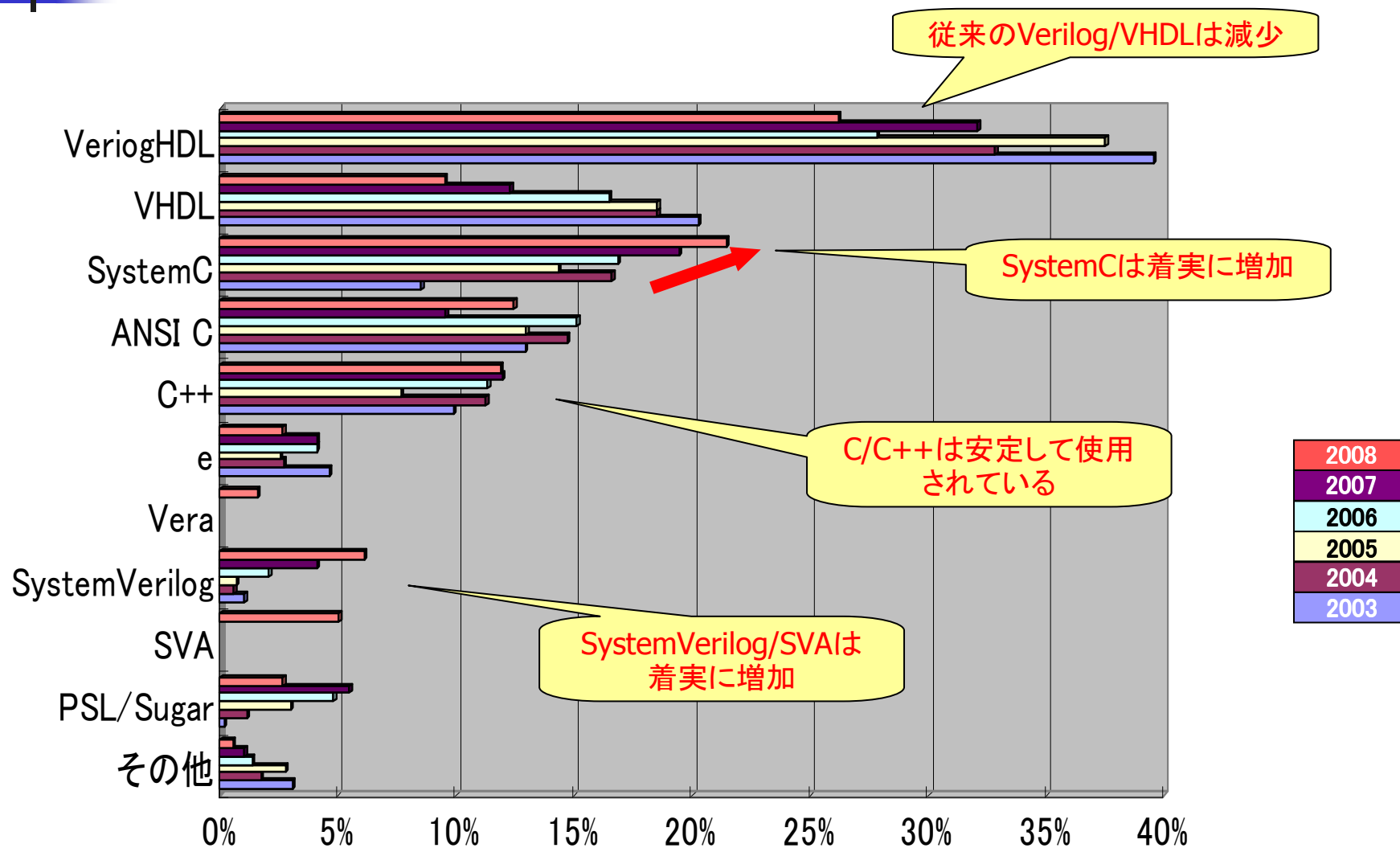
1. ご担当業務またはビジネスは?



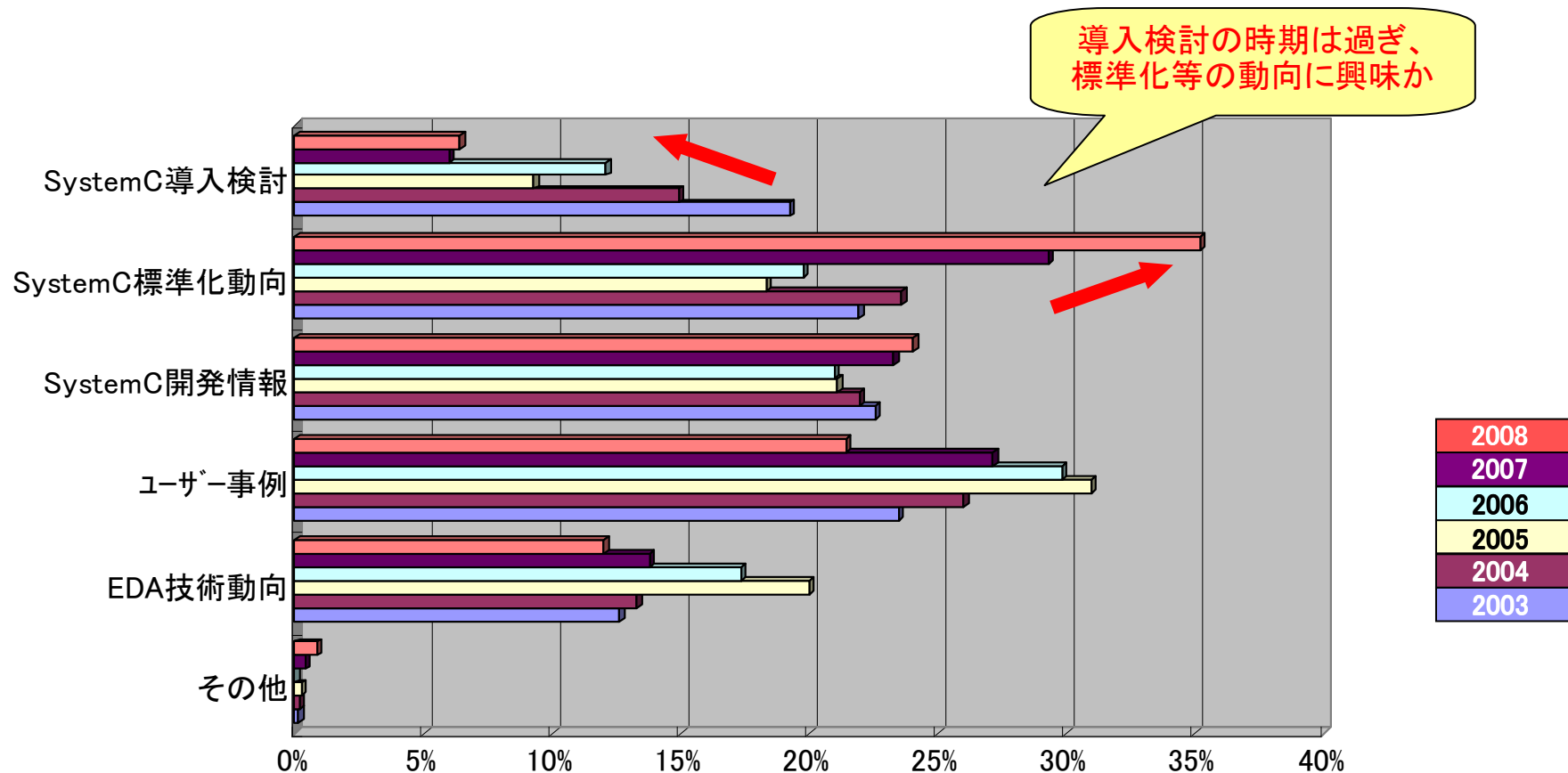
2. ご担当製品アプリケーションは？



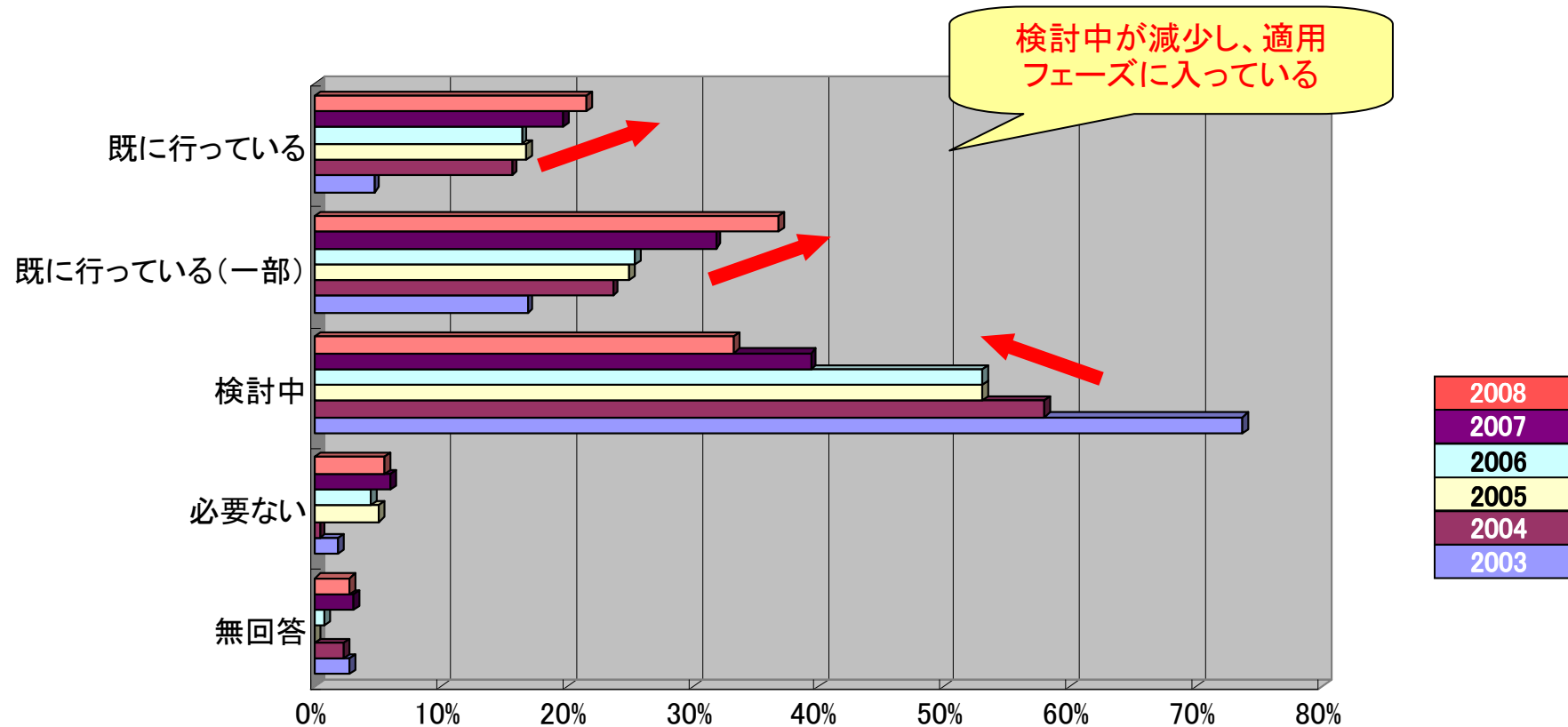
3. 現在主に使用している言語は?



4. SystemCユーザフォーラムに参加された目的は?

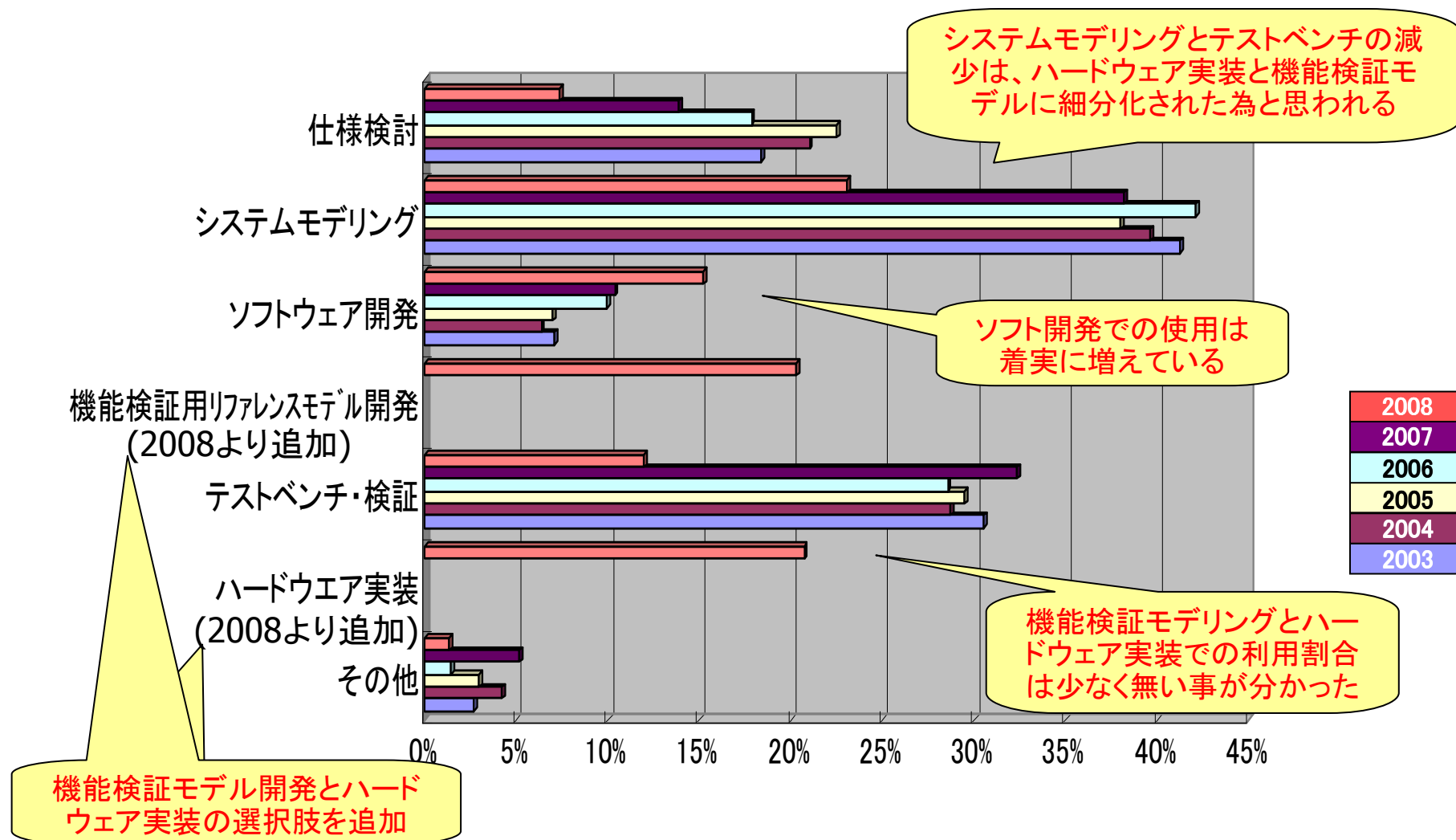


8. SystemCでの設計・検証環境構築について

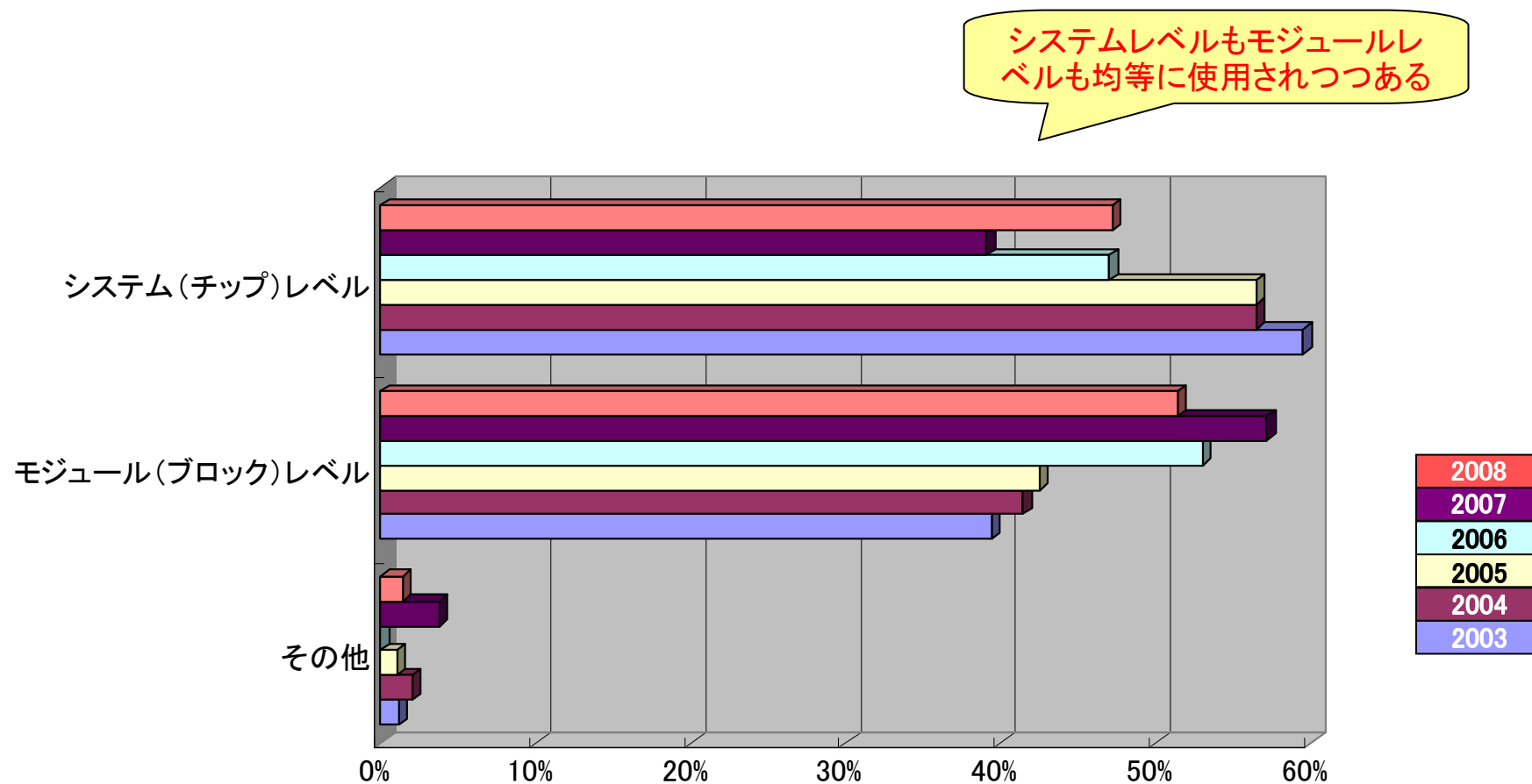


6. 「5」で「既に行っている」または「検討中」と回答された方

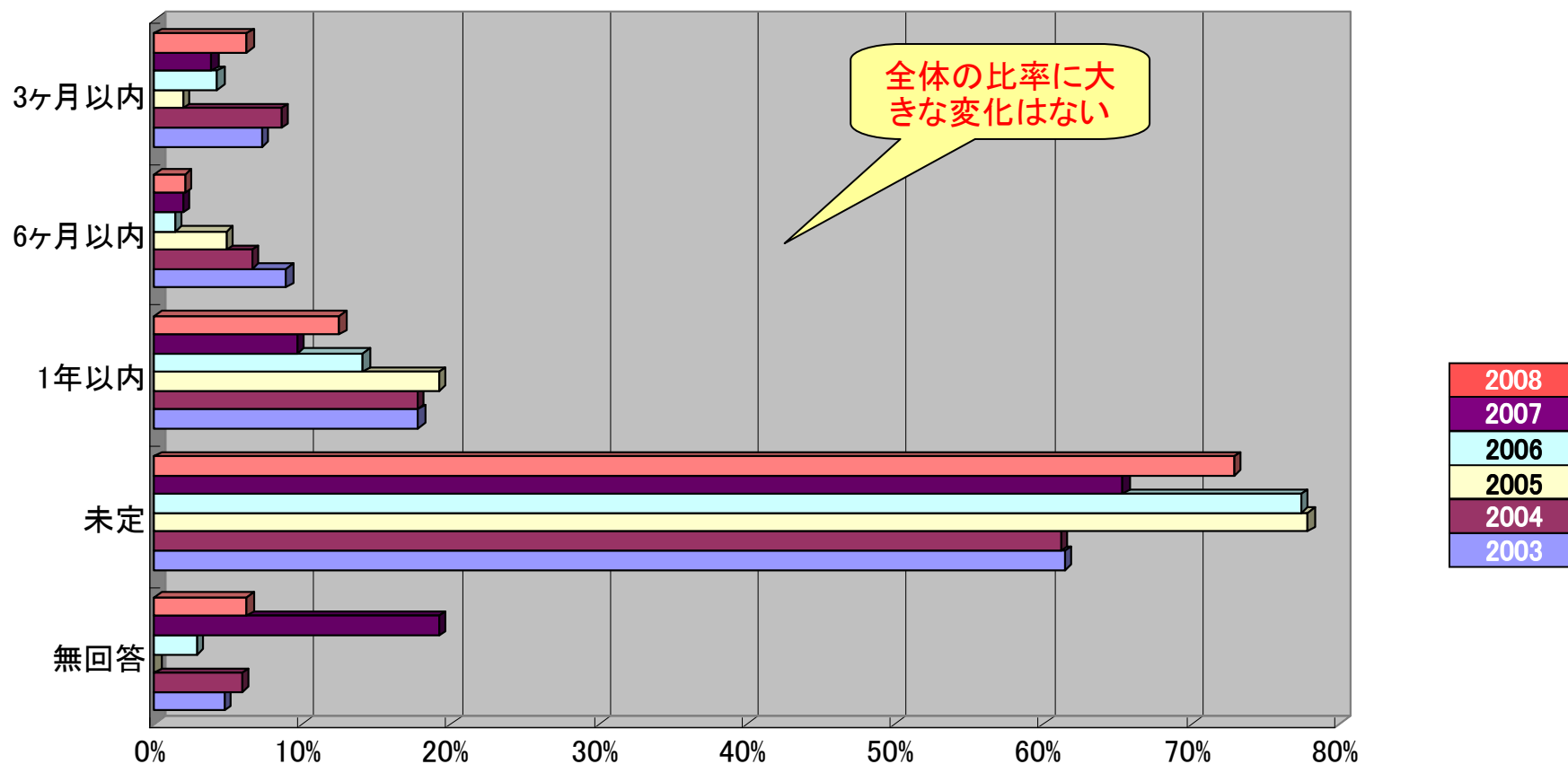
a) SystemCの使用目的は?



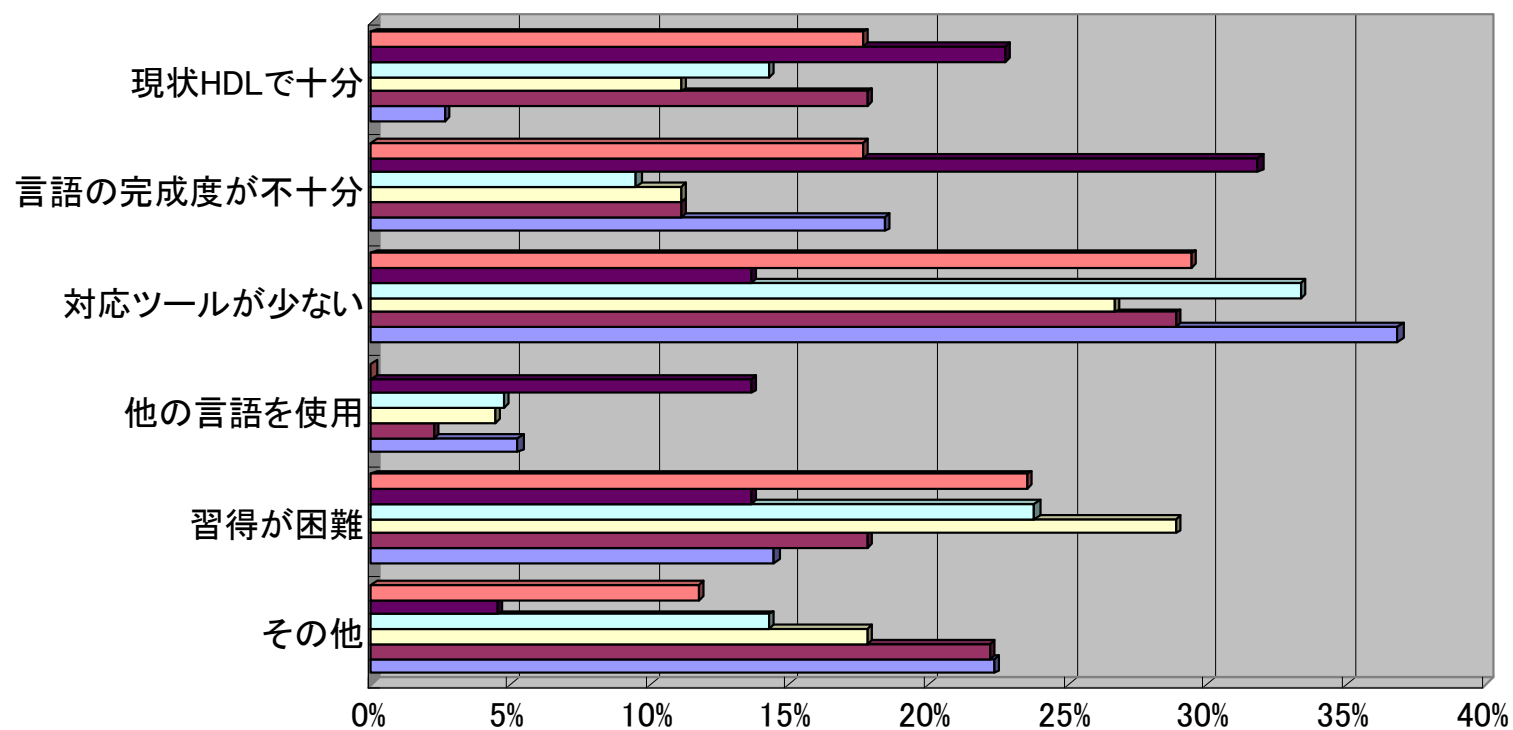
6. 「5」で「既に行っている」または「検討中」と回答された方 b) SystemCの活用範囲は?



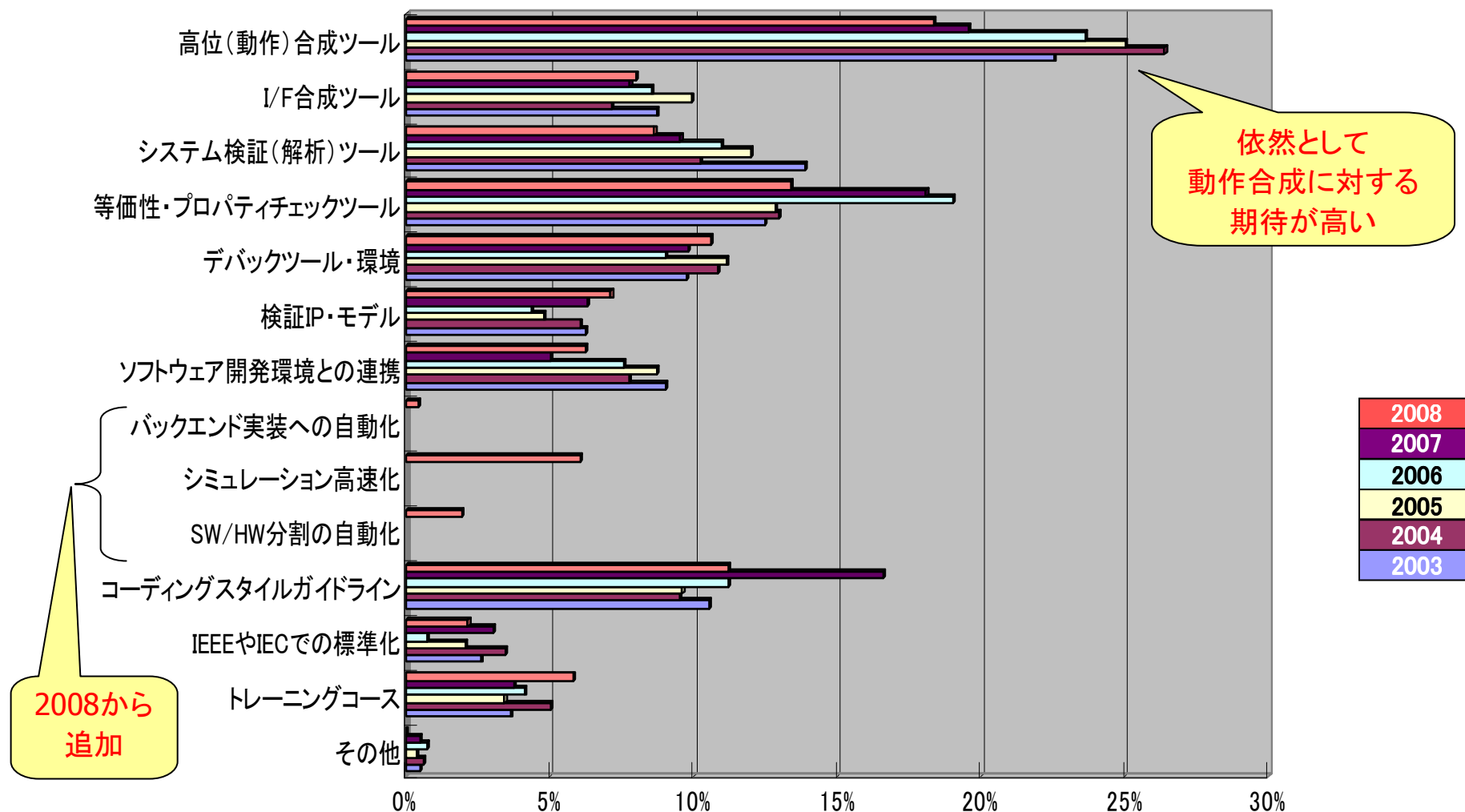
7. 「5」で「検討中」と回答された方へ 導入予定時期は?



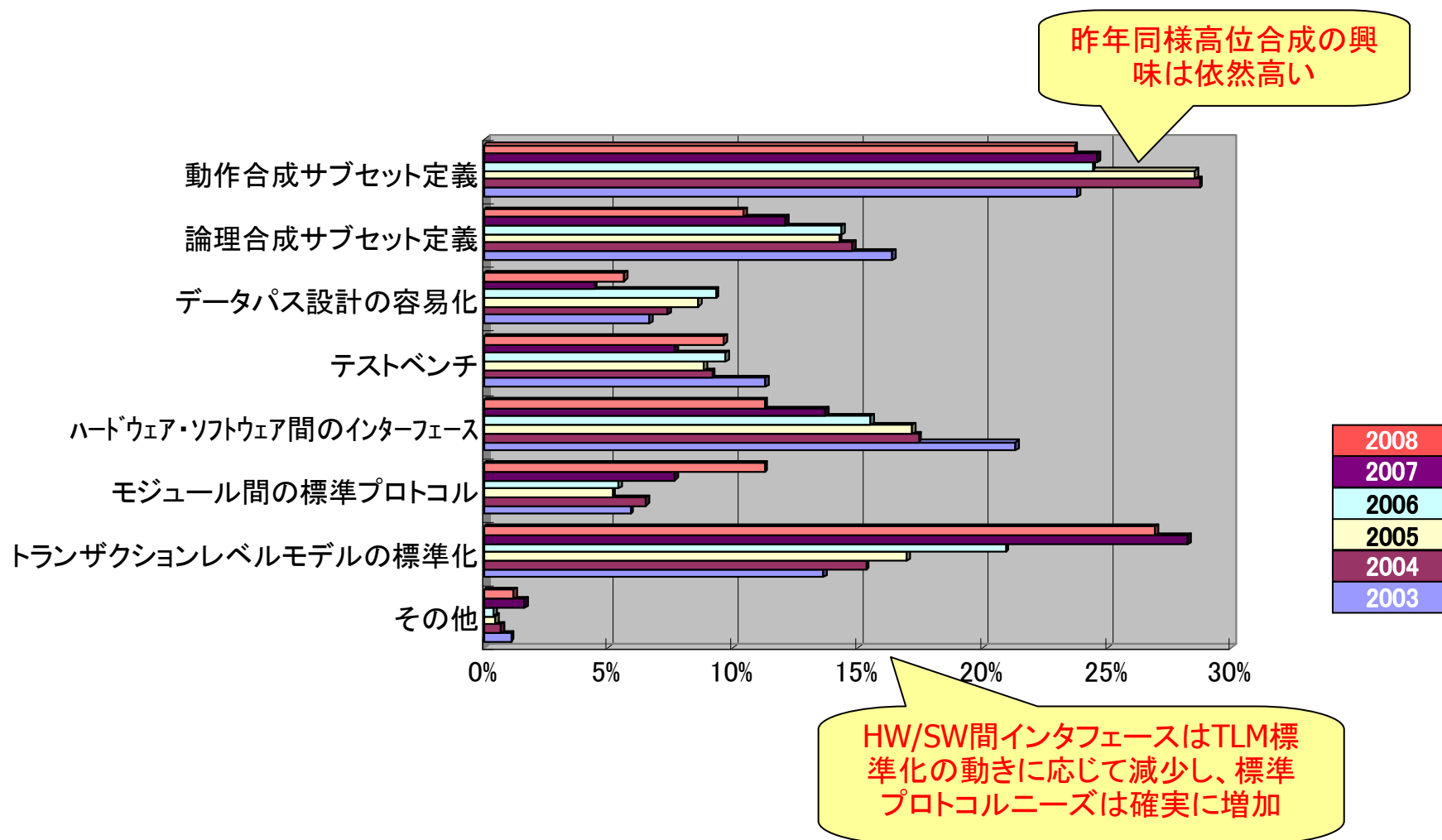
8. 「5」で「必要ない」、「検討中」と回答された方へ 導入の弊害となっている理由は?



9. SystemCをより活用する為に充実が必要なものは？



10. 今後SystemCの言語拡張・標準化で期待することは?



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

