何が嬉しい？何が変わった？MISRA-C：2012

Toyo Corporation
Software Solution

株式会社 東陽テクニカ

www.toyo.co.jp/ss/

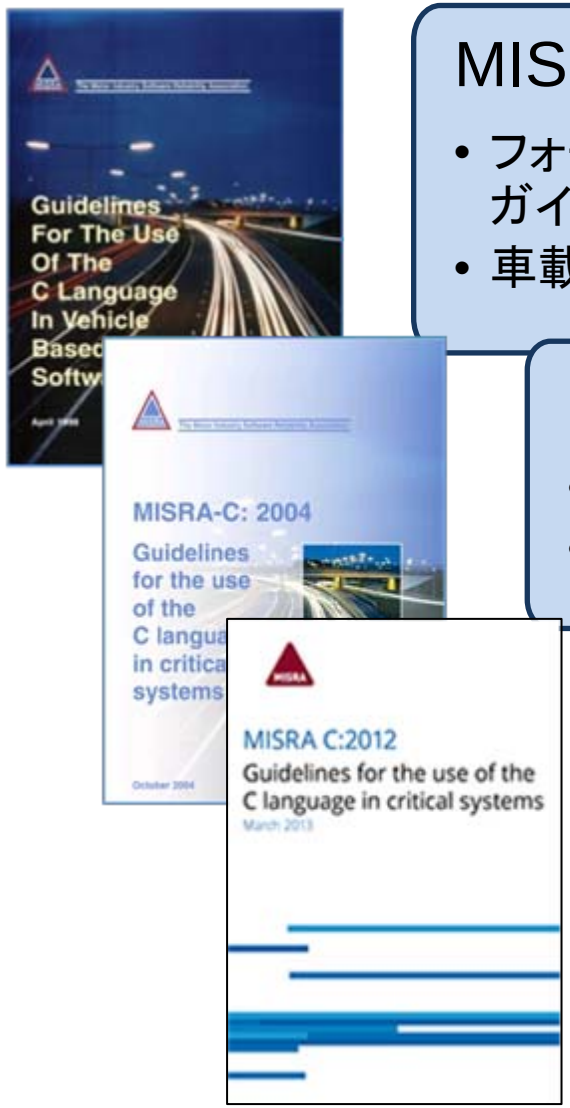
本日の内容

1. MISRA Cとは？
2. MISRA C:2012の発行経緯
3. MISRA C:2004とMISRA C:2012の主な違い
4. 既存コードへのMISRA C:2012の適用

本日の内容

1. MISRA Cとは？
2. MISRA C:2012の発行経緯
3. MISRA C:2004とMISRA C:2012の主な違い
4. 既存コードへのMISRA C:2012の適用

MISRA Cとは?



MISRA C:1998

- フォード社およびローバー社向けにPRQA社が開発したコーディングガイドラインがベースになっている
- 車載ソフトウェア向けのコーディングガイドラインとして、英国で策定

MISRA C:2004

- MISRA C:1998を改訂し、新しいルールを追加
- Exemplar suite によるルール解釈の補足

MISRA C:2012

- 経験豊富な10人のメンバーで構成されるMISRA委員会により策定
- 4年間におよぶ策定作業
- 2013年3月18日に発行
- MISRA C:2004からの大幅な進歩

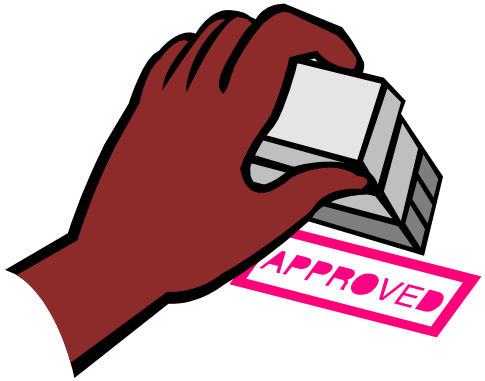
MISRA C:2004のルールは、
IPA/SECの中で多数参照されている

MISRA Cの意義

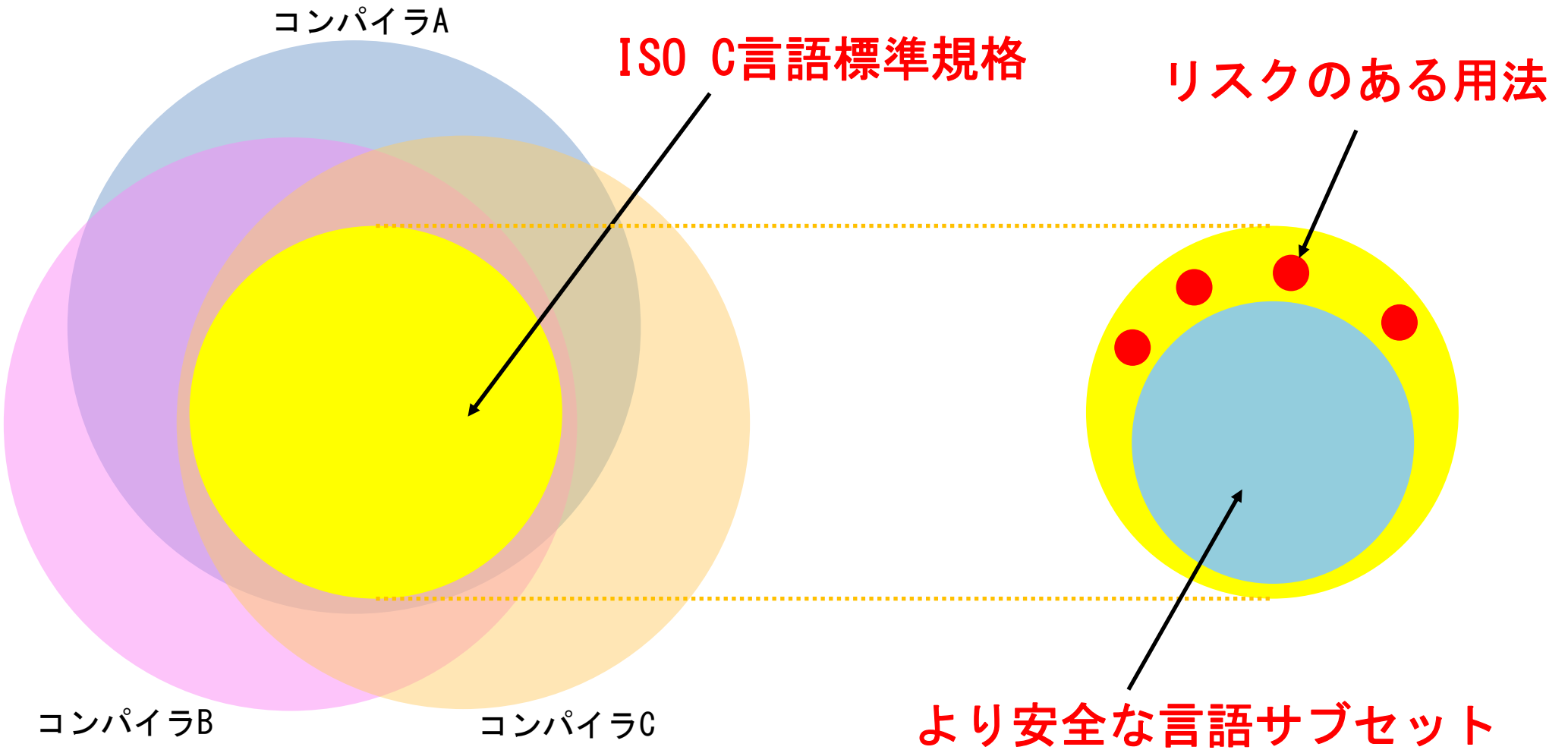
1. *より安全な言語サブセット* の定義

2. *形骸化* の回避

3. *コード品質* の改善
   - 信頼性
   - 保守性
   - 移植性



より安全な言語サブセットの定義



より安全な言語サブセットの定義

ISO 26262-6　5.4.7

設計及び実装の正しさを支援するために, モデリング又はプログラム言語に対する設計とコーディングのガイドラインは, 表1で列挙された項目を取り扱わなければならない。

IEC61508をベースにもつ
機能安全規格は同等の規定をもつ

表1の抜粋

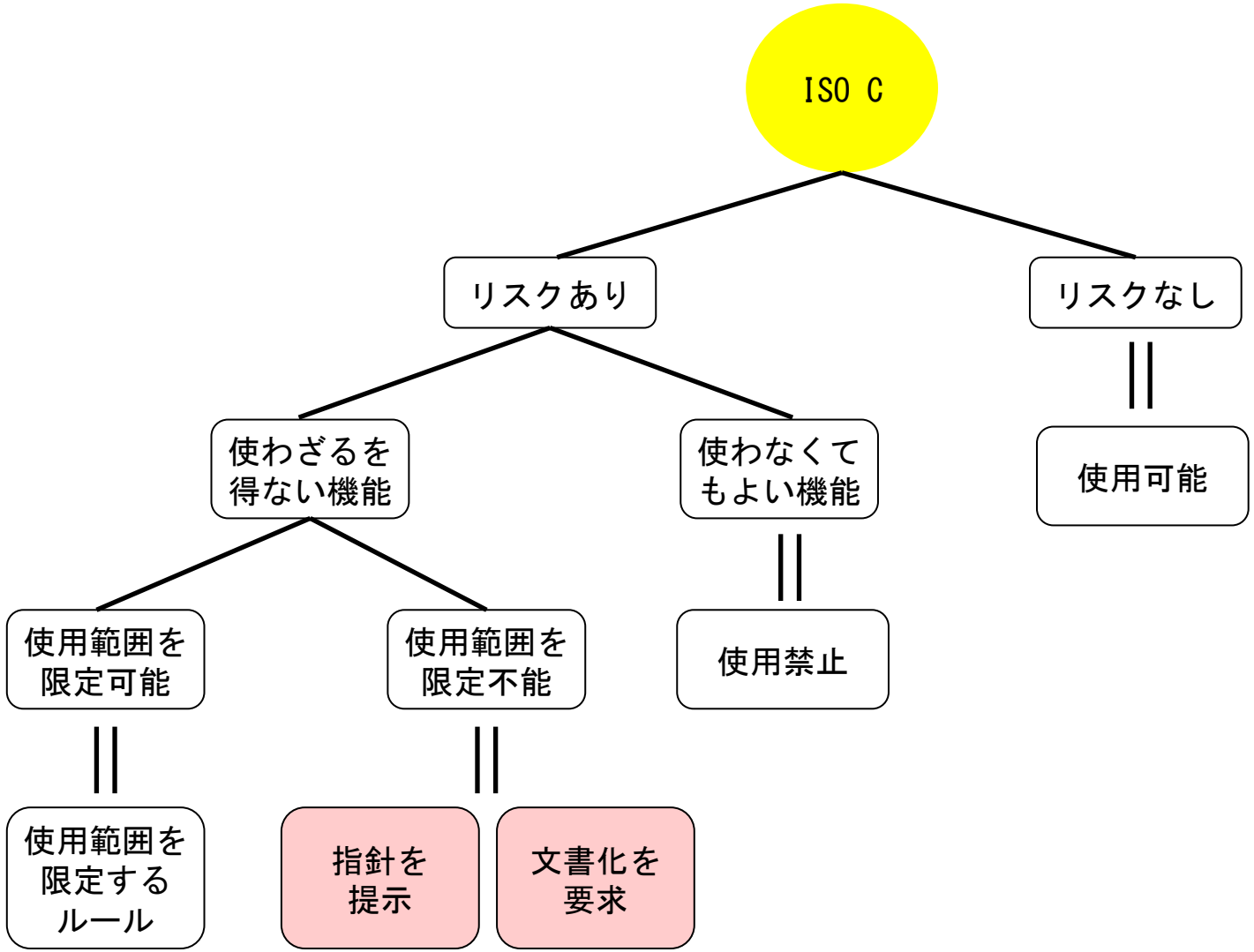
1b 言語サブセットの使用[b]

b 手法1bの目的は, 次のとおり。

- 異なるモデル作成者, プログラマ, コードジェネレータ, 又はコンパイラによって, 異なった解釈をされるかもしれない, 曖昧に規定された言語構文の排除.
- 経験上, 間違いを簡単に犯しやすい, 例えば, 条件付き代入又はローカル変数, 及びグローバル変数の同じ名前のような言語構文の排除。
- 処理されずに終わる可能性がある実行時エラーとなる言語構文の排除。

出典: INTERNATIONAL STANDARD ISO 26262-6 英和対訳版

リスクに基づいてルールを規定



MISRA Cが ＋α で注目したリスク

株式会社 東陽テクニカ

プログラミング言語にそもそも備わっている‘脆弱性’

未定義の動作 など

C99では 190個以上

文法的にも制約的にも問題はなく、コンパイラは
エラーを出力する義務も、マニュアルで注意を
促す義務も、動作を保証する義務もない

```
int buf[10];
...
buf[10] = 0;
```

C90 6.3 より
式の評価中に例外(exception)が発生する場合(すなわち、結果が数学的に定義できないか、又は結果の型で表現可能な値の範囲にない場合)、**その動作は未定義とする**。

形骸化の回避

株式会社 東陽テクニカ

ツールの力を最大限借りられる明確なルールを策定

例えば

MISRA C:2012

Rule 12.4 Evaluation of constant expressions should not lead to unsigned integer wrap-around
(定数式の評価では，ラップアラウンドが発生しないようにすべきである)
[MISRA C:2004 Rule 12.11]

リスクの対象を“変数”ではなく、“定数式”に限定することで
ツールの力を最大限借りられるルールを策定

本日の内容

1. MISRA Cとは？
2. **MISRA C:2012の発行経緯**
3. MISRA C:2004とMISRA C:2012の主な違い
4. 既存コードへのMISRA C:2012の適用

発行の動機

MISRA C:2004の利用者に与えていた過剰な負担を軽減する。

- 利便性の高いC99の機能を「逸脱の手続き」なしで利用できるように！
- ルール違反になる条件を明確にして、誤解釈を防止！
- 静的解析ツールの適用範囲を明確にして、遵守手段を整理！
- 「危険度の高い型変換」を検出するための枠組みを完成形に！
- 自動生成コードの扱いを一つのドキュメントに一本化！
- 厳しすぎて現実的ではなかったルールの見直し！

ISO C99対応

株式会社 東陽テクニカ

ISO:C90

- 様々なコンパイラおよびツールによる充実したサポート
- 危険性に関する十分な理解・浸透
- 制約 - 例: ブール型は使用できない

MISRA C:1998
MISRA C:2004
MISRA C:2012

ISO:C99

- _Bool型、inline関数など、より多くの機能が利用可能に
- 未定義の動作の追加などによる、危険性の増加
- C99のすべての機能をサポートするコンパイラは、ほとんど存在しない

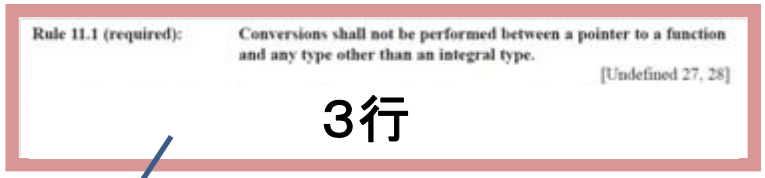
MISRA C:2012

ISO:C11

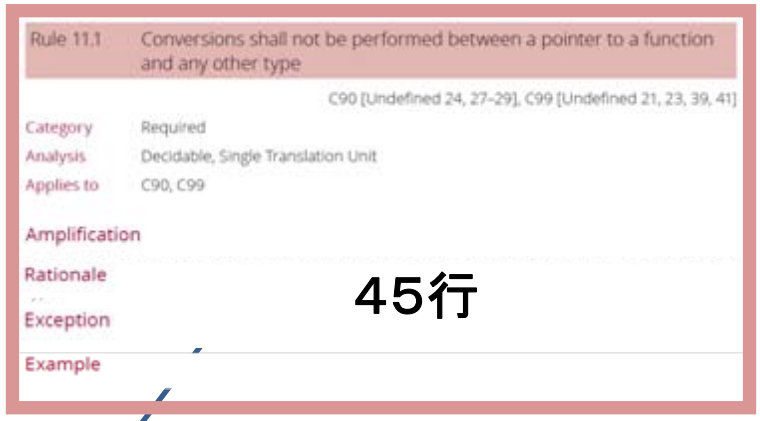
- 現在のところ最新のC言語標準規格
- C11をサポートしているツールは、非常に限られている

十分な適用実績をもつ
C99規格の機能を利用可能に！c

ガイドラインの内容改善



ガイドラインの要件が説明文の中に
紛れていることもあった



ガイドラインの構造化、
内容の充実、
内容の明瞭化、
不要な制約の除去を実施

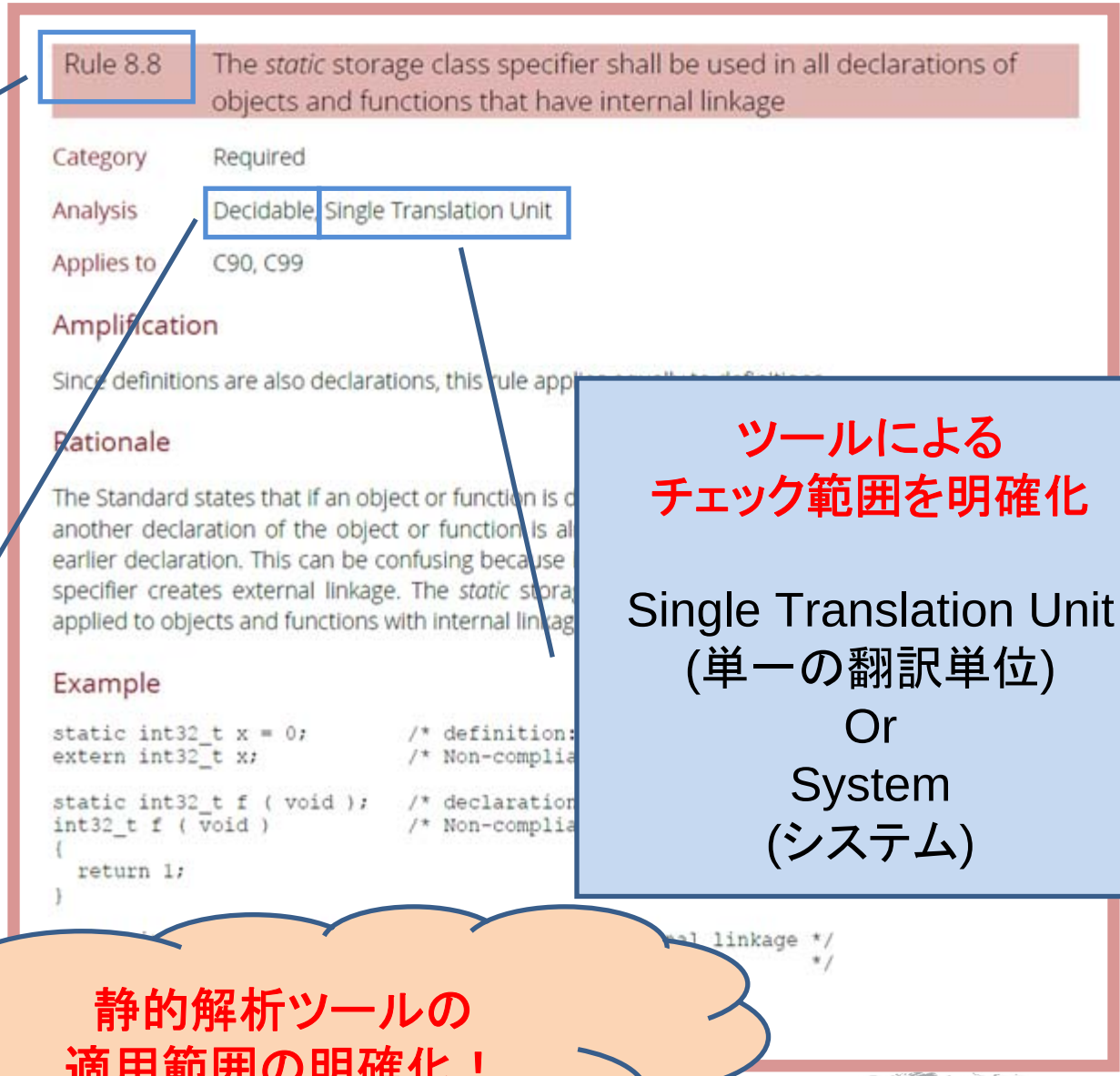
ルールに違反する条件の
明確化！

ツールの適用範囲の明確化

株式会社 東陽テクニカ

ガイドラインの分類を明確化

Rule (ルール)
Or
Directive (指針)



ツールによるチェック限界を明確化

Decidable (決定可能)
Or
Undecidable (決定不能)

ツールによるチェック範囲を明確化

Single Translation Unit
(単一の翻訳単位)
Or
System
(システム)

静的解析ツールの適用範囲の明確化！

ガイドラインの分類

株式会社 東陽テクニカ

MISRA C:2012から、ガイドラインは2つの種類に分類されています。

Rule (ルール)

- 要件が明確に定義されている
- 一部のルールを除いて静的にチェックすることが可能

Directive (指針)

- 要件が十分明確には定義されていない
  （要件の解釈に幅を持たせることが可能）
- “プロセス要件”または“文書化要件”に対応可能

静的解析ツールの適用を想定しているガイドラインが、ルールに分類されます。

チェック限界の明確化

株式会社 東陽テクニカ

MISRA C:2012から、すべてのルールが“Decidable (決定可能)” または “Undecidable (決定不能)”に分類されています。

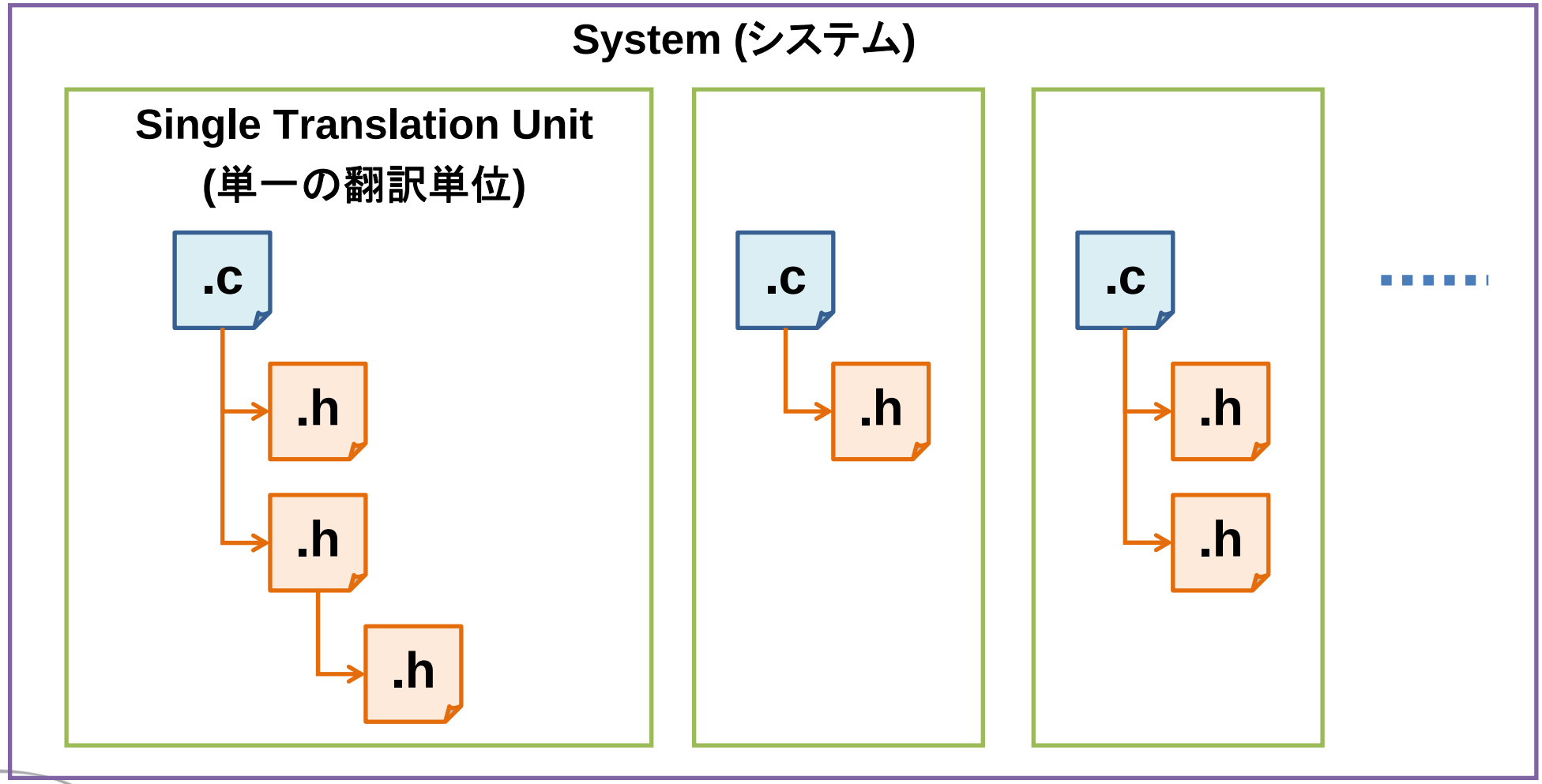
どのようなプログラムにおいても、コードがルールに準拠しているか否かをあるツールが*常に判断できる*場合、そのルールは*“Decidable (決定可能)”* とみなされます。それ以外の場合は、*“Undecidable (決定不能)”* とみなされます。

ルールに準拠しているか否かを論理的に区別できるツールを作り出せると考えられる場合、そのルールはDecidableに分類されています。

※
Decidableに分類されているルールであっても、そのルールがすべてのツールで確実にチェックできるとは限りません。

チェック範囲の明確化

MISRA C:2012から、すべてのルールが“System (システム)” または
“Single Translation Unit (単一の翻訳単位)”に分類されています。



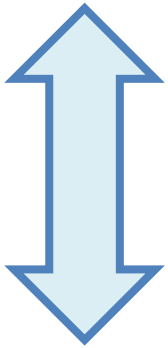
MISRA C独自の概念の改善

株式会社 東陽テクニカ

MISRA C:2004から危険度の高い型変換を“システマティック”に抽出するための新しい概念と用語を導入

```
signed long sl;
sl = 16384 + 16384 + 16384L;
     /* オーバーフロー （危険な型変換を含む式） */
```

※ int =16bit long = 32bitの場合



上の式は危険度高、下の式は危険度低と判断できるようにする

```
signed long sl;
sl = 16384L + 16384 + 16384;
     /* OK */
```

※ int =16bit long = 32bitの場合

MISRA C独自の概念の改善

MISRA C:2004	MISRA C:2012
underlying type (潜在型)	essential type (実質的な型)
complex expression (複合式)	composite expression (複合式)
effectively Boolean (実質的なブール型)	essentially Boolean (実質的なブール型)

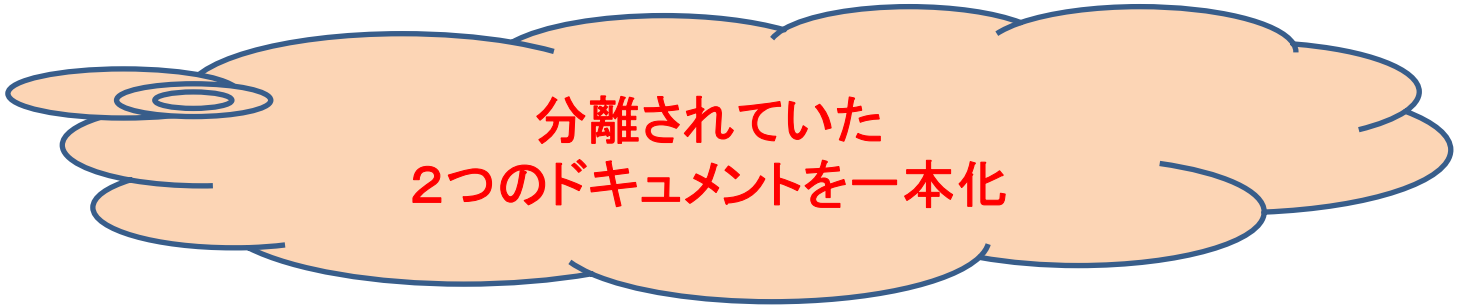
先進的な取り組みであったが、
列挙型やビットフィールドが
考慮されていないなどの
多くの穴があった

どのような算術型の式でも
表すことができる
完成された枠組みを構築

自動生成コードの扱い

自動生成コードに関する別ドキュメント“MISRA-AC-AGC”を統合するために以下の付録が追加されました。

Appendix E Applicability to automatically generated code



ルールの見直し

株式会社 東陽テクニカ

Rule	コード例	MISRA C 2004	MISRA C 2012
10.1	uint16_t ui = 42; // 右辺の方は signed int	違反	OK
2.2	x = y; // C++形式のコメント	違反	OK
19.4	#define MM { for (i=0; i<N; ++i) { foo();} } // do-whileでないマクロ	違反	OK
18.1	struct S *pst; // Sは不完全型	違反	OK

過剰な制限を課していた
ルールの見直し

本日の内容

1. MISRA Cとは？
2. MISRA C:2012の発行経緯
3. **MISRA C:2004とMISRA C:2012の主な違い**
4. 既存コードへのMISRA C:2012の適用

MISRA C:2012における変更

MISRA C:2012 をMISRA C:2004と比較すると…

- コンテンツ量の増加
  - 追加/削除/統廃合によって、ガイドラインの数が従来の142個から159個に変化
  - コンテンツを構造化して、より充実した内容に更新
- ガイドラインの多くの内容は実質的に変更されていない
  - 表現が見直され、ガイドラインの定義がより明確に
  - ガイドラインの番号は変更されている
- 従来のコードはMISRA C:2012に準拠しない場合がある
  - いくつかの新しい要件が追加されている
  - ただし、いくつかの制約が削除されている

追加/変更/削除に関する情報

株式会社 東陽テクニカ

MISRAの掲示板で、「マッピング情報」と「変更/削除内容」に関する資料を公開

MISRA C:2012	MISRA C:2004
Dir 4.13 (advisory)	New
Rule 1.1 (required)	Rule 1.1 (required)
Rule 1.2 (advisory)	Rule 1.1 (required)
	Rule 2.2 (required)

MISRA C:2004	MISRA C:2012	変更/削除内容
Rule 10.5 (required)	Deleted	This rule has been deleted as it did not adequately address the underlying problem. The enhanced type rules address some of the issues previously covered by this rule.
Rule 10.6 (required)	Rule 7.2 (required)	
Rule 11.1 (required)	Rule 11.1 (required)	Tightened to include conversions to/from integral types. Relaxed to permit conversions from a *null pointer constant*, and conversion to *void*.

コンテンツの構造化

株式会社 東陽テクニカ

要件を
明確化

Amplification
要件の詳細な説明

Rationale
ガイドラインの必要
性に関する説明

Example
より多くの
コード例を追加

Exception
例外事項を追加

Rule 8.8 The *static* storage class specifier shall be used in all declarations of objects and functions that have internal linkage

Category	Required
Analysis	Decidable, Single Translation Unit
Applies to	C90, C99

分類を追加

Amplification

Since definitions are also declarations, this rule applies equally to definitions.

Rationale

The Standard states that if an object or function is declared with the *extern* storage class specifier and another declaration of the object or function is already visible, the linkage is that specified by the earlier declaration. This can be confusing because it might be expected that the *extern* storage class specifier creates external linkage. The *static* storage class specifier shall therefore be consistently applied to objects and functions with internal linkage.

Example

```
static int32_t x = 0;         /* definition: internal linkage  */
extern int32_t x;             /* Non-compliant                 */

static int32_t f ( void );    /* declaration: internal linkage */
int32_t f ( void )            /* Non-compliant                 */
{
  return 1;
}

static int32_t g ( void );    /* declaration: internal linkage */
extern int32_t g ( void )     /* Non-compliant                 */
{
  return 1;
}
```

コンテンツの構造化

株式会社 東陽テクニカ

Amplification, Rationale, Exception, Exampleを追加

項目	目的
Amplification	ガイドラインの詳細な要件を記述して、ガイドラインの見出し文が複雑になることと、要件がRationaleに紛れてしまうことを回避
Rationale	今まで通り、ガイドラインの必要性を説明
Exception	ガイドラインの要件から除外する事項を記述して、ガイドラインの見出し文の複雑になることと、要件がRationaleに紛れてしまうことを回避
Example	準拠／非準拠を明確に示したコード例を提示

MISRA C:2004

Rule 11.1 (required): Conversions shall not be performed between a pointer to a function and any type other than an integral type. [Undefined 27, 28]

3行

MISRA C:2012

Rule 11.1 Conversions shall not be performed between a pointer to a function and any other type

C90 [Undefined 24, 27-29], C99 [Undefined 21, 23, 39, 41]

Category Required

Analysis Decidable, Single Translation Unit

Applies to C90, C99

Amplification

Rationale

Exception

Example

45行

Amplificationの例

株式会社 東陽テクニカ

MISRA C:2004

Rule 8.11 The *static* storage class specifier shall be used in definitions and declarations of objects and functions that have internal linkage
（static 記憶域クラス指定子は，内部結合をもつオブジェクト及び関数の定義及び宣言に対して用いなければならない）

「定義」は「宣言」を兼ねる用語であり、冗長な記述であったため、ガイドラインの見出しから分離してAmplificationに記述

MISRA C:2012

Rule 8.8	The *static* storage class specifier shall be used in all declarations of objects and functions that have internal linkage
Amplification	Since definitions are also declarations, this rule applies equally to definitions.

Exceptionの例

株式会社 東陽テクニカ

MISRA C:2004

Rule 7.1　Octal constants (other than zero) and octal escape sequences shall not be used.
((0以外の)8進定数及び8進拡張表記は，用いてはならない)

「(0以外の)」の部分をガイドラインの見出しから分離してExceptionに記述

※ 8進拡張表記に関する要件はRule 4.1に分離

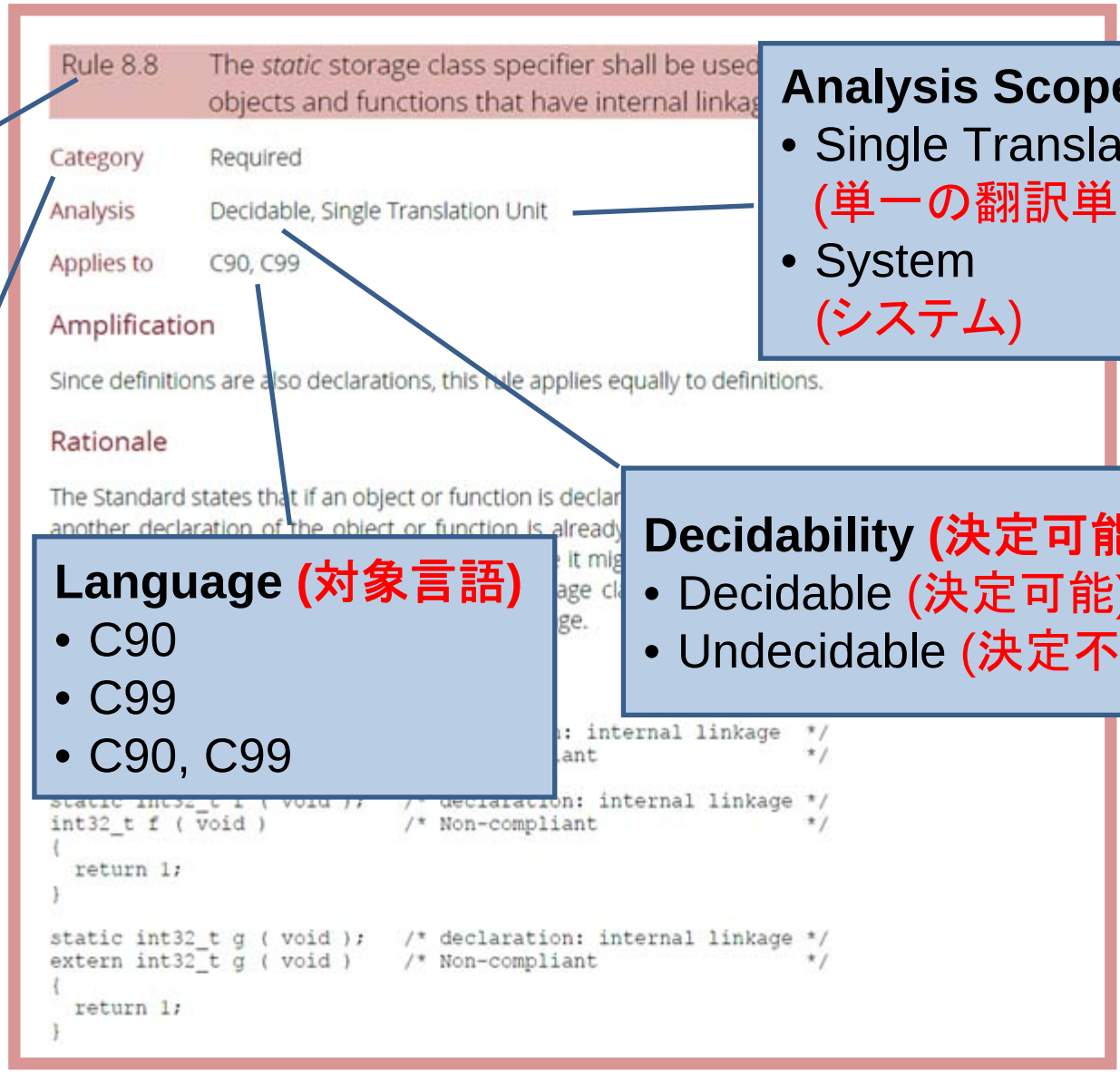
MISRA C:2012

Rule 7.1　Octal constants shall not be used
Exception The integer constant zero (written as a single numeric digit), is strictly speaking an octal constant, but is a permitted exception to this rule

ガイドラインの分類

Type (種類)
- Directive(指針)
- Rule（ルール）

Category (カテゴリ)
- Advisory（推奨）
- Required（必要）
- Mandatory（必須）



Analysis Scope (解析範囲)
- Single Translation Unit (単一の翻訳単位)
- System (システム)

Language (対象言語)
- C90
- C99
- C90, C99

Decidability (決定可能性)
- Decidable (決定可能)
- Undecidable (決定不能)

ガイドラインの分類に関するまとめ

株式会社 東陽テクニカ

分類のまとめ		Directive (指針) (16)	Rule (ルール) (143)
Category (カテゴリ)	Advisory (推奨)	7	32
	Required (必要)	9	101
	Mandatory (必須)	0	10
Language (対象言語)	C90	0	2
	C99	0	11
	C90 または C99	16	130
Decidability (決定可能性)	Decidable (決定可能)	-	117
	Undecidable (決定不能)	-	26
Analysis Scope (解析範囲)	Single Translation Unit (単一の翻訳単位)	-	104
	System (システム)	-	39

MISRA C:2012から、いくつかのルールが“Mandatory(必須)"に分類されました

いくつかのルールは、特定のC90とC99のどちらかにのみ対応します

ルールを静的にチェックした場合に、いくつかのルールに関しては、その確実性が担保されません

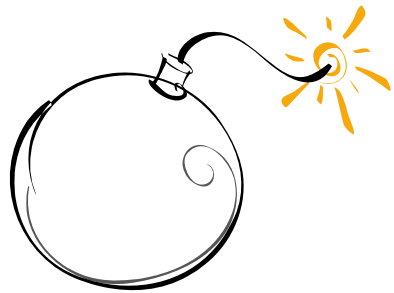
単一の翻訳単位内でチェックできるルールは、決定可能なルールです

本日の内容

1. MISRA Cとは？
2. MISRA C:2012の発行経緯
3. MISRA C:2004とMISRA C:2012の主な違い
4. **既存コードへのMISRA C:2012の適用**

既存コードへのMISRA C:2012の適用

新たに追加したコードと変更を加えたコードにのみ
MISRA C:2012の適用を検討するべきです。



レガシーコードは一定水準の品質が担保されている
とみなして、MISRA C:2012の適用対象外とするべきです。

MISRA C:2012の適用にあたって

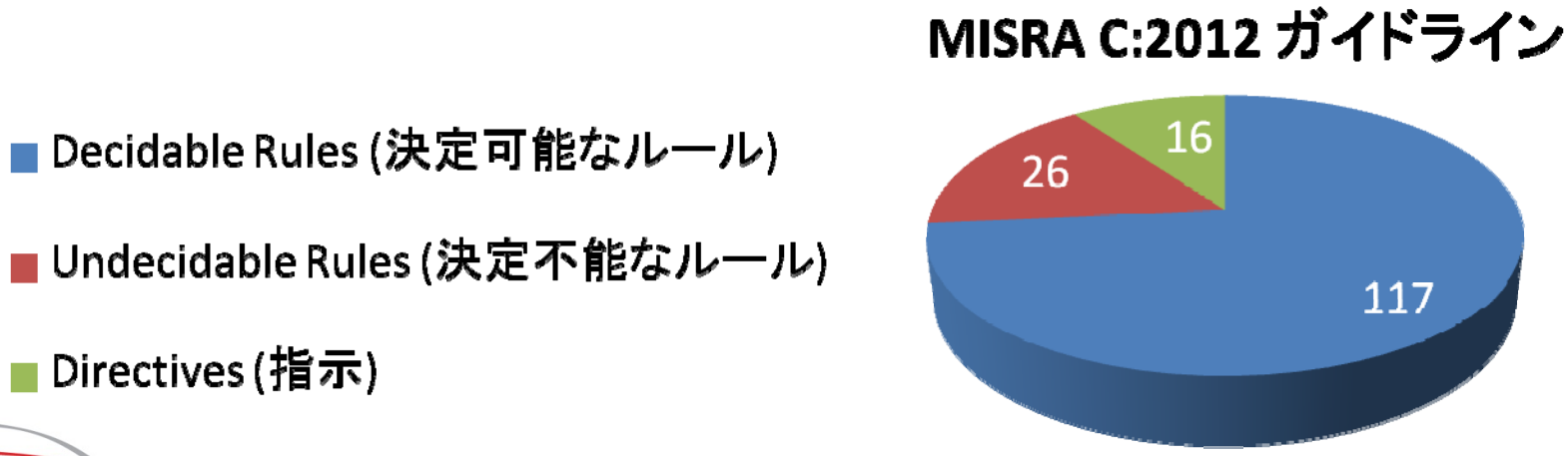
MISRA C:2012 Section 5.2.1 - Process activities required by MISRA C

"In order to use MISRA C, it is necessary to develop and document …
“MISRA Cを適用する際は、以下のものを作成・文書化する必要があります…

- *A compliance matrix, showing how compliance with each MISRA C guideline will be checked“*
- *それぞれのMISRA Cガイドラインのチェック方法を示した準拠マトリクス*

MISRA C:2012 Section 5.3 - Compliance

"Where a guideline cannot be completely checked by a tool, then a manual review will be required.“
“ツールでチェックできないガイドラインは、マニュアルレビューを実施する必要があります。”



逸脱の手続き

MISRA C:2012 Section 5.2.1 - Process activities required by MISRA C

"In order to use MISRA C, it is necessary to develop and document …

“MISRA Cを適用する際は、以下のものを作成・文書化する必要があります…

- *A deviation process by which justifiable non-compliances can be authorized and recorded “*
- *非準拠箇所の妥当性を承認および記録できる逸脱の手続き*

MISRA C:2012 Section 5.4 – Deviation procedure

" It is important that such deviations are properly recorded and authorized.“

“このような逸脱の手続きは適切に承認および記録することが重要です。”

逸脱の手続きは、時として避けることができません。
しかしながら、逸脱の手続きを乱用してしまうと、
プロセスの完全性が損なわれます。

MISRA C ADC: Approved deviation compliance for MISRA C:2004
ISBN 978-906400-09-5 (PDF), February 2013.
www.misra.org.uk

MISRA C:2012関連情報

MISRA C:2012の概要資料と ホワイトペーパーのダウンロード

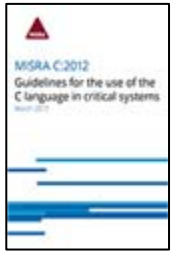
PRQA社のMISRA C:2012サイト: www.programmingresearch.com/mc3/

- 概要資料(英語):
  「Fact Sheet about MISRA C3」のリンクをクリックして下さい。
- ホワイトペーパー(英語):
  「White Papers」のフォームにご登録下さい。

東陽テクニカの会員サイト: www.toyo.co.jp/ss/intro_members/

- 概要資料(和訳) :
  会員登録後に、
  「ホワイトペーパー (東陽作成)」>「FACT SHEET - MISRA C:2012」
  の順でリンクをクリックして下さい。
- ホワイトペーパー(和訳) :
  会員登録後に、
  「ホワイトペーパー (メーカー作成)」>「ホワイトペーパー申込み: MISRA C:2012」
  の順でリンクをクリックして下さい。

MISRA C:2012関連情報



MISRA C:2012の入手

MISRAのサイト: www.misra.org.uk/Buyonline/tabid/58/Default.aspx

- 「The MISRA webstore」のリンクをクリックして下さい。

MISRA C:ADCとMISRA C:2004との比較資料の入手

MISRAの掲示板: www.misra.org.uk/Discussions/tabid/61/Default.aspx

- 「The MISRA Bulletin Board」のリンクをクリックして下さい。

本セミナーの内容と重複する点もありますが、
MISRA C:2012に関するWebinarを6月に開催いたします。
http://www.toyo.co.jp/ss/webinar/

質疑応答