



## An Introduction to MISRA C++

Chris Tapp, MISRA C++ Chairman  
([chris.tapp@ldra.com](mailto:chris.tapp@ldra.com))

Copyright © 2009 Liverpool Data Research Associates Limited

- MISRA C++ ワーキンググループ
- MISRA C++ 背後にある根拠
- 主な機能
- 効果
- 組込み、安全性システムへの適用
- 今後の予定
- まとめ

- MISRA-C++ ワーキンググループの全メンバーは自費負担のボランティア
- 中心メンバー
  - Richard Corden, Programming Research
  - Mike Hennell, LDRA
  - Derek Jones, Knowledge Software
  - Clive Pygott, QinetiQ
  - Chris Tapp, LDRA (Chairman)
- MIRA はアドミニストレーションサービスを提供 for MISRA
- 連絡先
  - web: [www.misra-cpp.com](http://www.misra-cpp.com)
  - e-mail: [chairman@misra-cpp.org](mailto:chairman@misra-cpp.org)

- C++言語のスタンダード 脆弱性に関して
  - 指定されない振舞い
  - 未定義の振舞い
  - コンパイラにより指定される振舞い
  - 診断の必要のない振舞い
- C++は複雑な言語
  - 複雑度がプログラマに理解されにくい
  - メンテナンス性が困難になり得る
  - 隠蔽されるコード(コンストラクタなど)理解される必要がある
  - 想定外のタイプ変換/選択が起こり得る
- 標準となるサブセットがない
  - インハウスのスタンダードが使用されている
    - プロジェクトに依存してしまいがち
    - 多くの場合スタイルにこだわりをみせるが、これらは安全性には寄与しない

- 課題の検証・識別
  - ISO/IEC 9899:1990 ©とは異なり、ISO/IEC 14882:2003 (C++)は、未定義、指定されない、実行時の振舞いのリストがない
  - QinetiQの“脆弱性レポート”はISO/IEC 14882:2003のそのような全ての課題を列挙し明らかにした
- 既存のベストプラクティスを参照・採用
  - 他のコーディングスタンダード
    - MISRA-C
    - JSF++ (MISRA-C をベースにした)
    - 医療関係
    - 輸送、交通システム
  - ツールベンダとして現実的な経験から
  - その他
    - Scott Meyers
    - Stephen Dewhurst
    - Etc.

- 新しいルールの追加
  - 既存ベストプラクティスでカバーされていない問題を緩和するため
    - テンプレート
    - インヘリタンス、継承
    - エクセプション、例外
    - 不要なコンストラクト
  - ベストプラクティスを拡張するため
- ピア・レビュー 専門家による意見、評価
  - 最初にルールは開発プロセスに参与する組織によりレビュー
  - ドラフト(草案)を幅広く外部のレビュアーに
  - フィードバックは解析され、必要に応じてドキュメントの改訂

- ISO/IEC 14882:2003 C++ をターゲットに
  - ルールはISO/IEC 14882 のポジション番号にあわせてグループ化(クロスリファレンスしやすいように)
- MISRA Cに沿って開発
  - 全ルールに理論的根拠を定める
  - 全ルールに事例を定める(適切な場合に)
  - ルールは3つのカテゴリーに
    - Required –必須( MISRA C 同様)
    - Advisory –勧告、注意( MISRA C 同様)
    - Document –立証(文書で証明) - コンパイラやターゲットハードウェアなどでの確認が必要
- 全てのタイプのクリティカルシステムに当てはまる
- 殆どのルールはツールによりサポート
  - 検査実施コストの削減
  - プログラマー、レビューヤーの負荷を軽減
- ドキュメント化ルールによる移植の支援

- C++は組込みや安全システムには向かないと考えられてきたが、すでに広く用いられている
  - 医療
  - 航空宇宙 (e.g. Rolls-Royce, JSF)
  - 防衛 (e.g. Windows software used to determine required runway lengths for cargo flights)
  - ハイレベルな制御 (e.g. process control GUI)
- スキルが期待できることが主なる原動力
  - JSFでは40,000ものプログラマーが関わる。ADA言語(防衛関連などで必須であった)にした場合、そのような人数を確保できない
- 自動車関連も既に採用
  - テレマティクス
  - インフォテインメント(ナビなど)

- 見本集
  - 更なる例を用いてルールに対する追加の説明
  - ツール性能評価の支援
- C++ language
  - さらなる言語コアへのガイダンス(コードの完全性)
  - STL(Standard Template Libraries )や他のライブラリに対して
  - 新しいC++言語バージョン(new version of ISO 14882 )
- 国際的な協調作業
  - 日本の組織とも

- LDRA tool suite®
  - MISRA C++:2008 (released on 5 June 2008)
    - 228 ルール
    - 現在 75% に対応済み
  - LDRA はMISRA C++委員会を代表し、そのリリース日にサポート



- MISRA C++ではクリティカルシステムへのC++のベストプラクティスを規定する
  - あらゆるタイプのクリティカルシステムに該当
  - C++プログラマを対象とし、言語のエキスパートではない
  - MISRA C 同様に広く採用されることを願う
- LDRAツールスイートは
  - ルールの施行
  - コードレビュー、クオリティレビュー
  - 多様な標準への準拠

The screenshot displays the LDRA TBvision 7.0.2 interface. The main window shows a code review for 'System\_Integration\_Tbdemo' using the 'CERT Model Used' security selection. A table of violations is visible, with a red arrow pointing to a specific entry.

Violation	File	Line Number
Pointer not checked for null before use : match	Tbdem2.c	7
Pointer not checked for null before use : match	Tbdem2.c	9

The log window at the bottom shows the following messages:

```

Code Review Started - System_Integration_Tbdemo : with the C - MISRA
standards model(s) implemented
Code Review Completed
Code Review Started - System_Integration_Tbdemo : with the C - CERT
standards model(s) implemented
    
```

# LDRAツール MISRA-C++:2008 表示例/ コールグラフと



The screenshot displays the LDRA Toolchain 7.8.2 interface. The main window shows a list of MISRA-C++:2008 violations for the file 'Inheritance.cpp'. The violations table is as follows:

Number	Level of Violation	Phase Code	Standard Code
S 543	Required	S 543	MISRA-C++:2008 10-01-02
S 214	Required	S 214	MISRA-C++:2008 10-03-02
S 521	Advisory	S 521	MISRA-C++:2008 10-05-01
D 65	Required	D 65	MISRA-C++:2008 00-01-28
S 559	Required	S 559	MISRA-C++:2008 10-03-01
D 46	Required	D 46	MISRA-C++:2008 00-03-08
D 65	Required	D 65	MISRA-C++:2008 00-01-06
D 46	Required	D 46	MISRA-C++:2008 00-03-00

The 'Static Call Graph of program: Inheritance.tgt.10' window shows a hierarchical call graph starting from 'main'. The graph includes nodes for 'Visitor::Describe', 'Person::Describe', 'Person::Person', 'Employee::Describe', 'Employee::Employee', 'Student::Describe', 'Student::Student', and 'Trainee::Describe'. Red arrows indicate the flow of control between these functions.

The 'Division Log Window' at the bottom shows the following log entries:

```

LDRA Division 7.8.2 © 1975 - 2008 LDRA Ltd.
Processing File C:\LDRA_WORKAREA\Examples\Inheritance.tgt
Processing File Completed
Code Review Started - Inheritance.cpp - C:\LDRA_WORKAREA\Examples\Inheritance.tgt
with the C++ MISRA C++:2008 standards model(s) implemented
Code Review Completed
    
```