欲しくなるテストツールの機能とは

FUJI SETSUBI

FUJI SETSUBI 電子機器事業部

＊LDRA/テスト自動化ツール

静的解析、カバレッジ、単体テスト

＊T-VEC/モデル検証・テスト自動化

要求仕様～デザインモデル、実コードまで

＊MetaEdit+

Domain Specific Modeling 開発環境

＊Ashling/Jtag デバッガ

ARC、SmartCards等各種

＊ローターバッハ/Jtag デバッガ

ARM、PowerPC、68K、MIPS等各種

＊XJTAG

バウンダリスキャン・HWテストツール

- ソースコードベース
- モデル検証
- 自動コード生成（Domain-Specific Modeling）

- ソースコードベース
- モデル検証
- 自動コード生成

（Domain-Specific Modeling）

高信頼性向け
30年以上の実績

静的テスト

静的テスト
単体テスト

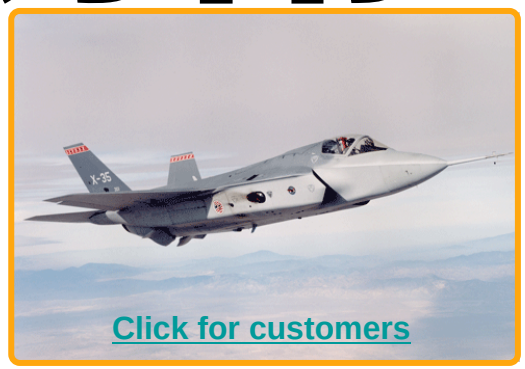
全て融合

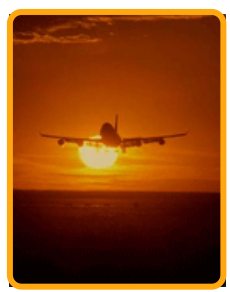
LDRA
Software Technology

静的テスト
単体テスト
テストカバレッジ

航空、防衛

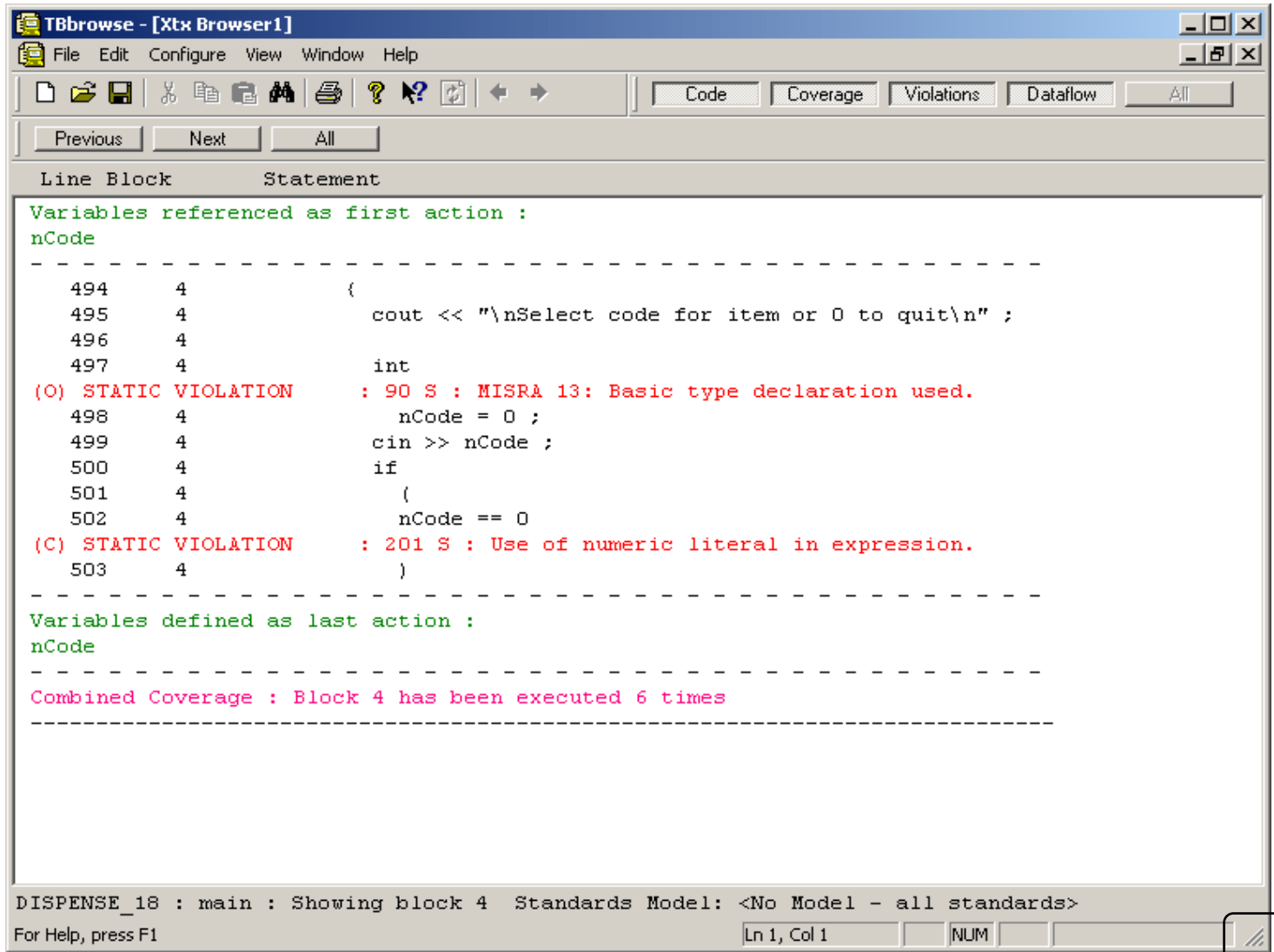
NORTHROP GRUMMAN
Litton Litton Systems
LOCKHEED MARTIN
BOEING
TRW
HUGHES
DUNLOP Aviation
OFFICINE GALILEO
CRANE AEROSPACE & ELECTRONICS
Ultra ELECTRONICS
Honeywell
MTU Aero Engines
SENER
KAISER ELECTRONICS
Rolls-Royce
GB TECH
smiths
Kidde Aerospace
Penny&Giles DRIVES TECHNOLOGY A SPIRENT Company
Bell Helicopter TEXTRON
AIRBUS INDUSTRIE
THALES Thales Avionics
BAE SYSTEMS





Back

MISRA 委員会

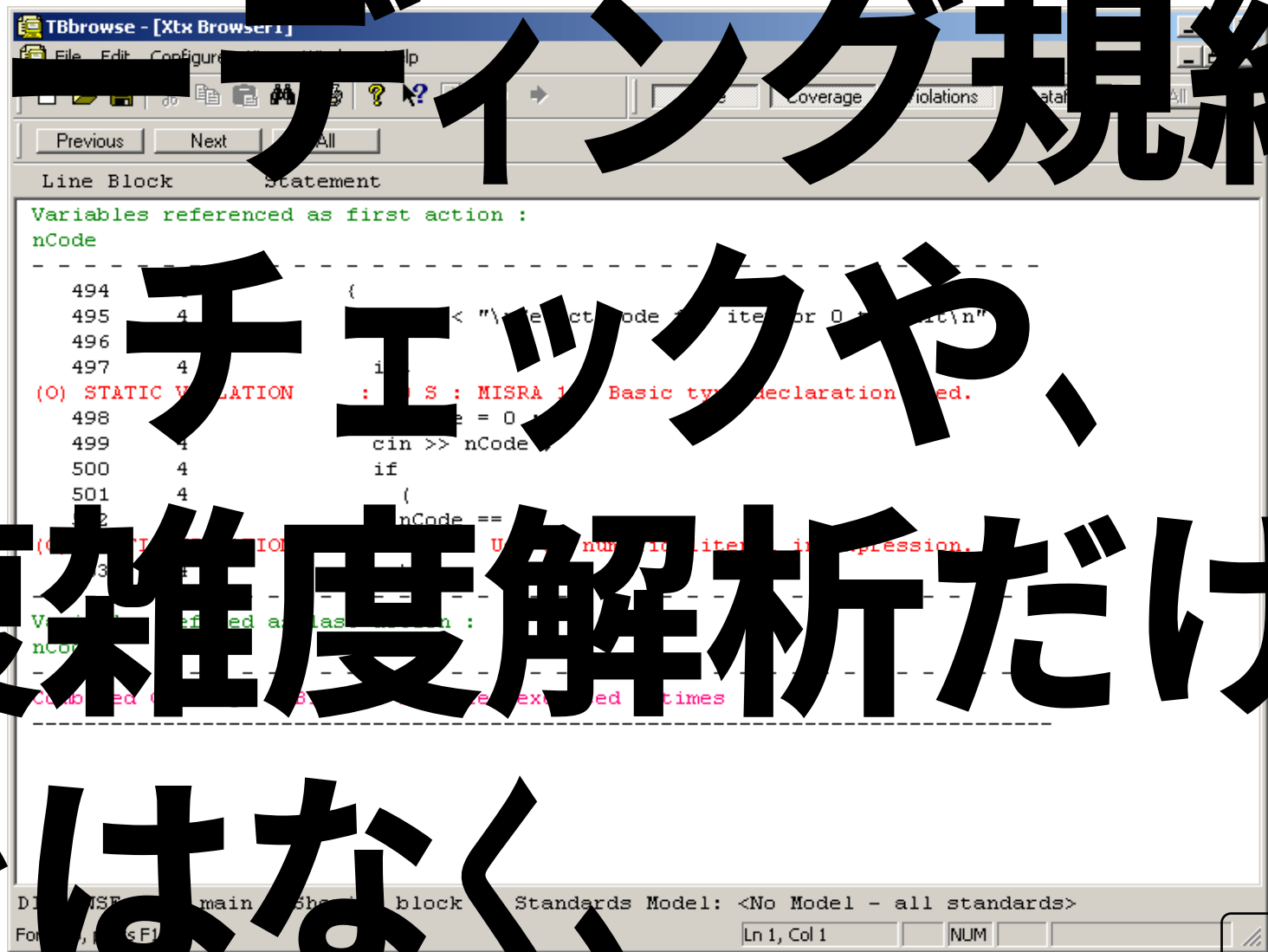


Back

の静的テストは

、

MISRA など コーディング規約チェックや、複雑度解析だけではなく、

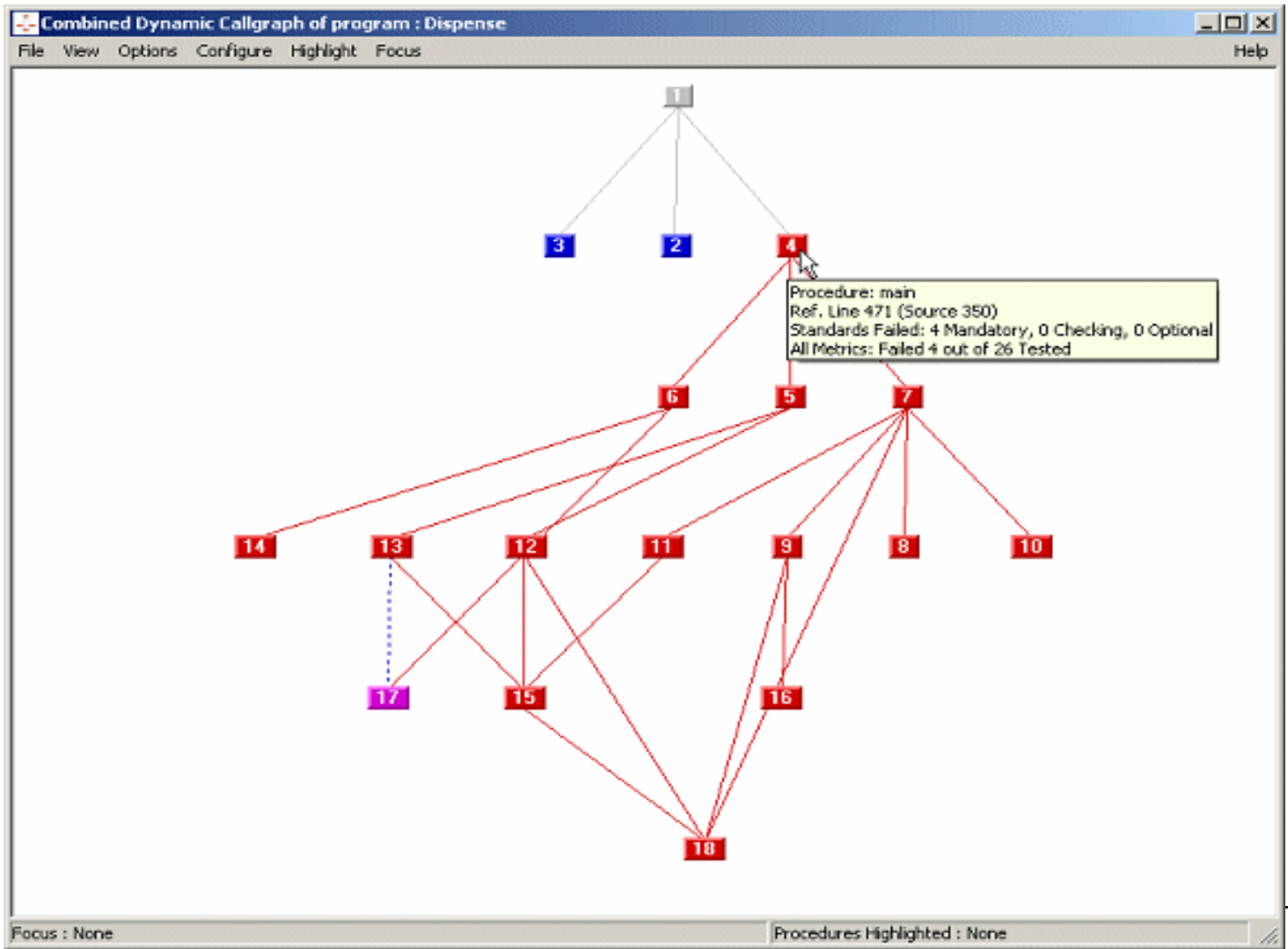


Back

視覚的に

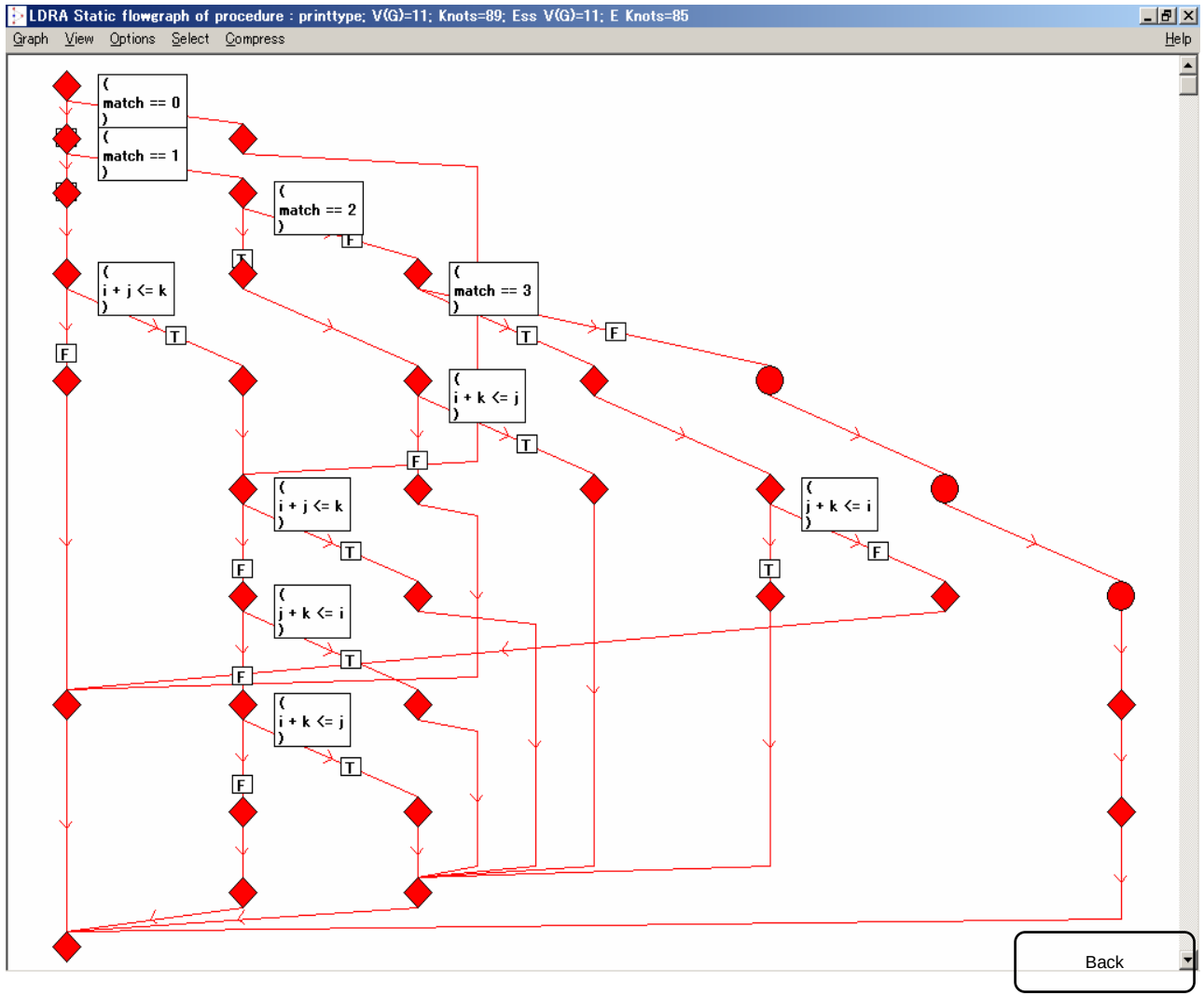
Back

関数コールや、

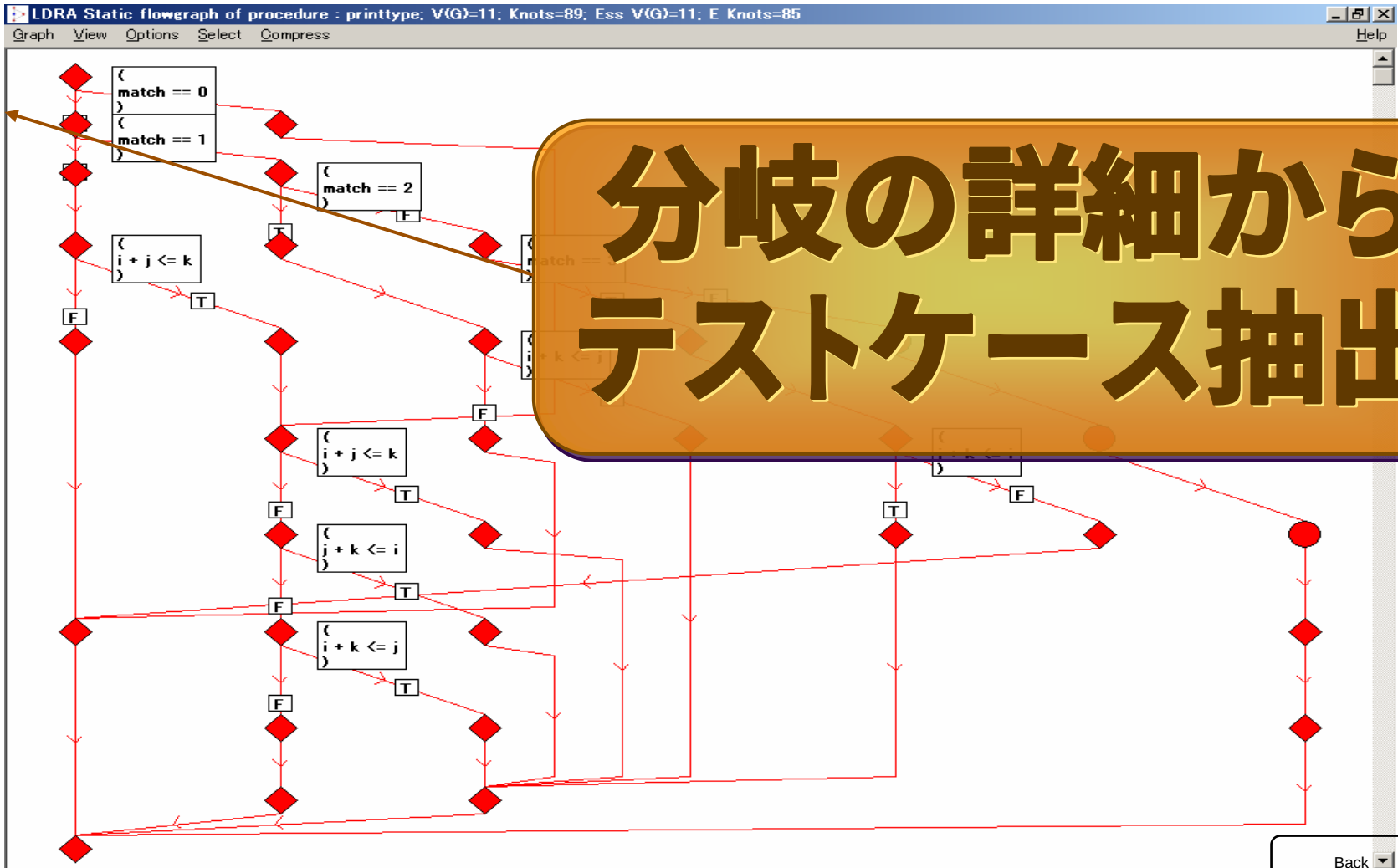

Combined Dynamic Callgraph of program : Dispense
File View Options Configure Highlight Focus Help
1
3
2
4
Procedure: main
Ref. Line 471 (Source 350)
Standards Failed: 4 Mandatory, 0 Checking, 0 Optional
All Metrics: Failed 4 out of 26 Tested
6
5
7
14
13
12
11
9
8
10
17
15
16
18
Focus : None
Procedures Highlighted : None

Back

実行フローを解析

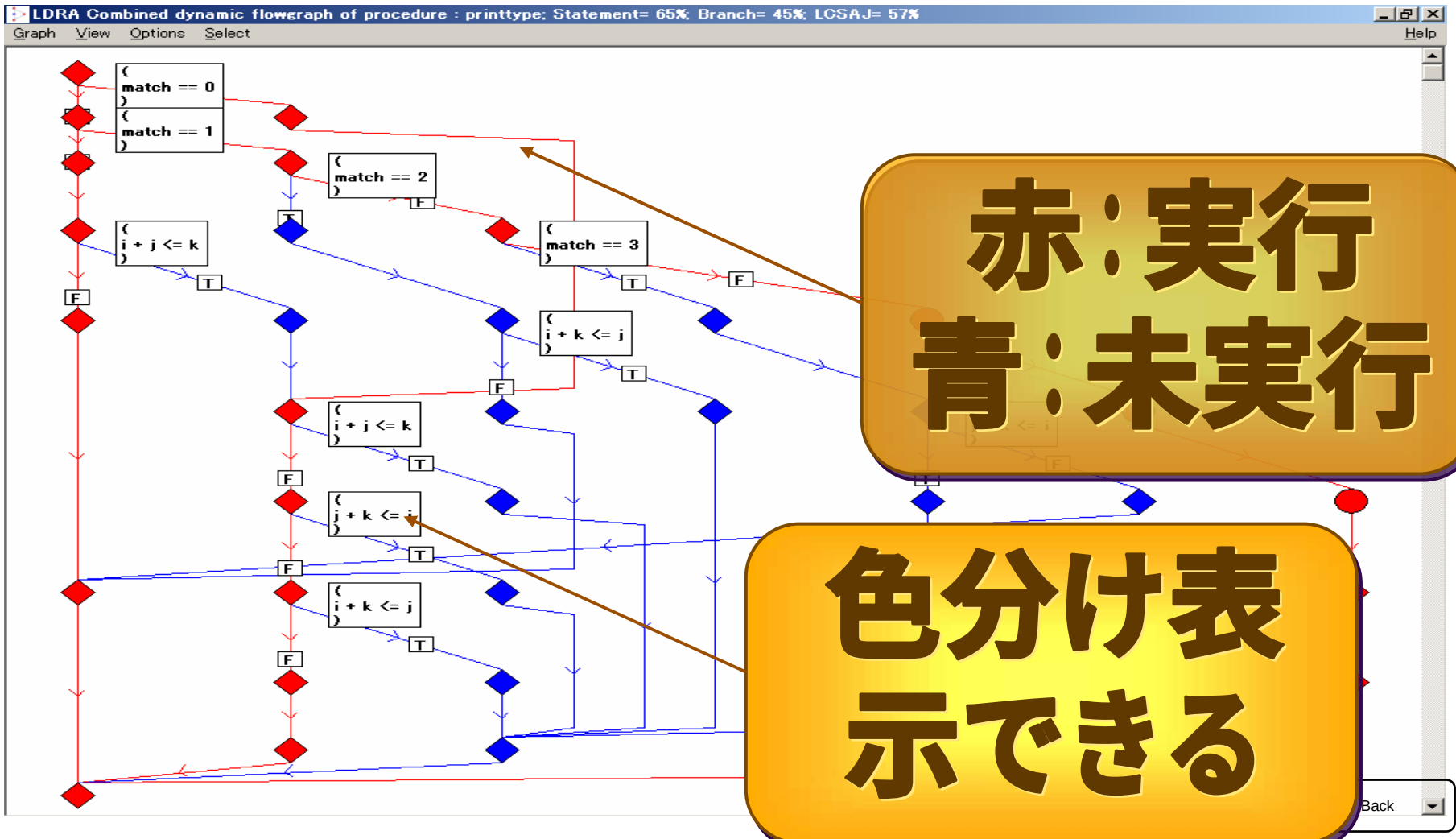

LDRA Static flowgraph of procedure : printtype; V(G)=11; Knots=89; Ess V(G)=11; E Knots=85
Graph View Options Select Compress Help
(match == 0)
(match == 1)
(match == 2)
(match == 3)
(i + j <= k)
(i + k <= j)
(j + k <= i)
T
F
Back

実行フロー内の



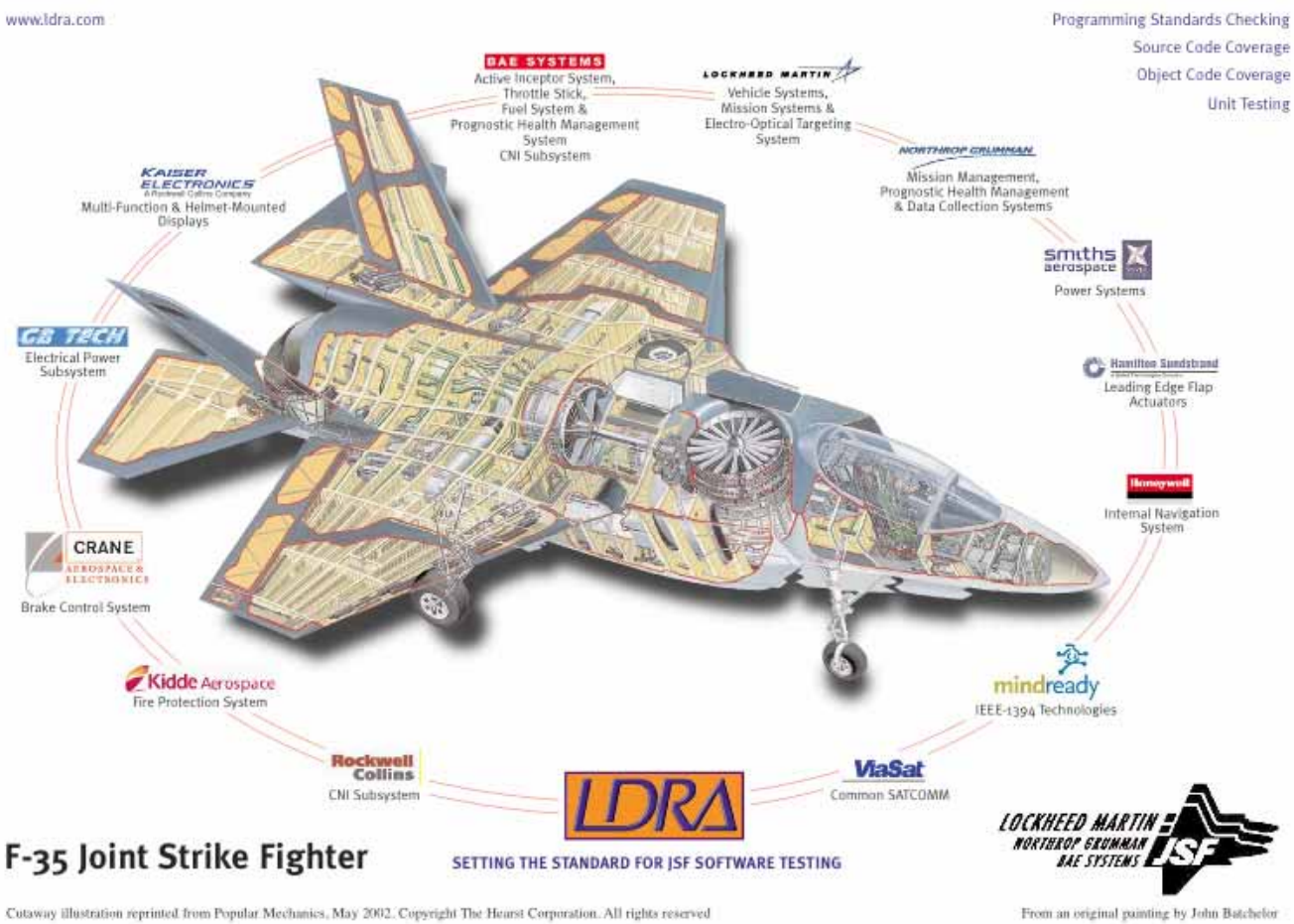
分岐の詳細から テストケース抽出

結果のカバレッジ


LDRA Combined dynamic flowgraph of procedure : printtype; Statement= 65%; Branch= 45%; LCSAJ= 57%
Graph View Options Select Help
match == 0
match == 1
match == 2
match == 3
i + j <= k
i + k <= j
j + k <= i
T
F
赤:実行
青:未実行
色分け表示できる
Back

静的テスト、単体テスト、カバレッジが有機的に融合される

JSF開発では、



Back

JSF開発では、

２０００～３０００人の

SW エンジニアの

標準ツールとして、

Back

JSF開発では、

コーディング規約

単体テスト

ソースコードカバレッジ

オブジェクトコードカバレッジ

Back

JOINT STRIKE FIGHTER

AIR VEHICLE

C++ CODING STANDARDS

FOR THE SYSTEM DEVELOPMENT AND DEMONSTRATION PROGRAM

Document Number 2RDU00001 Rev C

December 2005

JSF C++ コーディング規約

Back

静的テスト、
単体テスト、
カバレッジ

が有機的に融合されることで、

検証作業、

リグレション、

要求対テストのトレーサビリティ　を効率

DO-178B **全レベル**

MISRA C

MISRA-C:2004

IEC 61508 **全レベル**

高信頼性向けの機能、性能を持っていながら

大規模開発にも利用できうるコストパフォーマンスでご提供。

本数割引

2本目8割引

3本目以降9割引

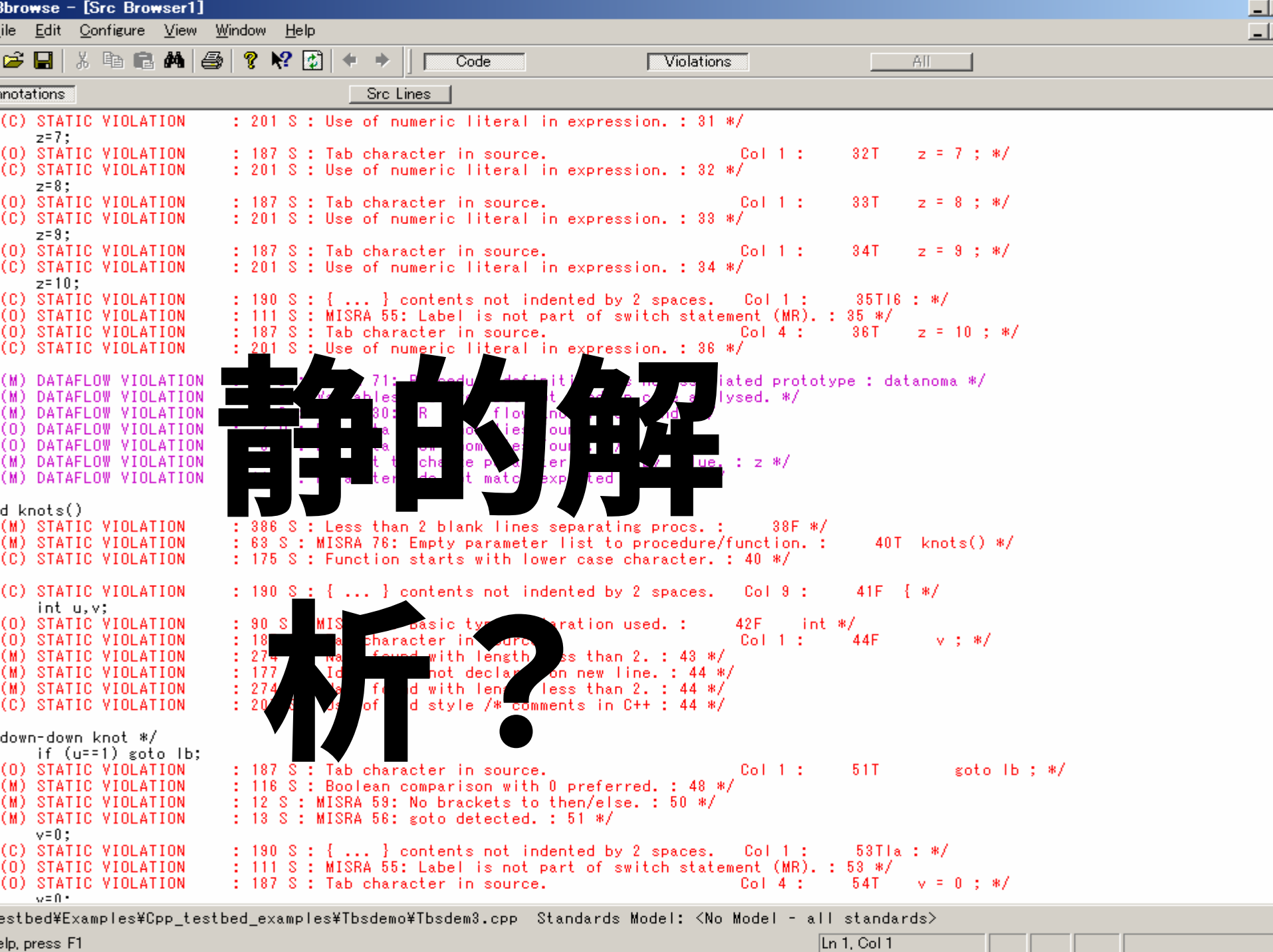
付帯条件なし

ところで、

高信頼性向けでは無い場合

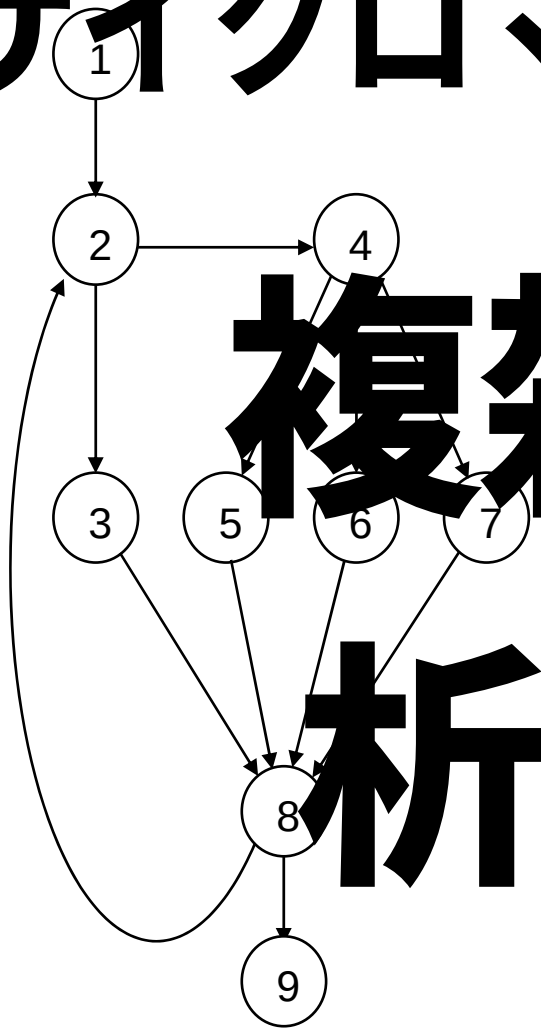
Back

静的解析


静的解析?

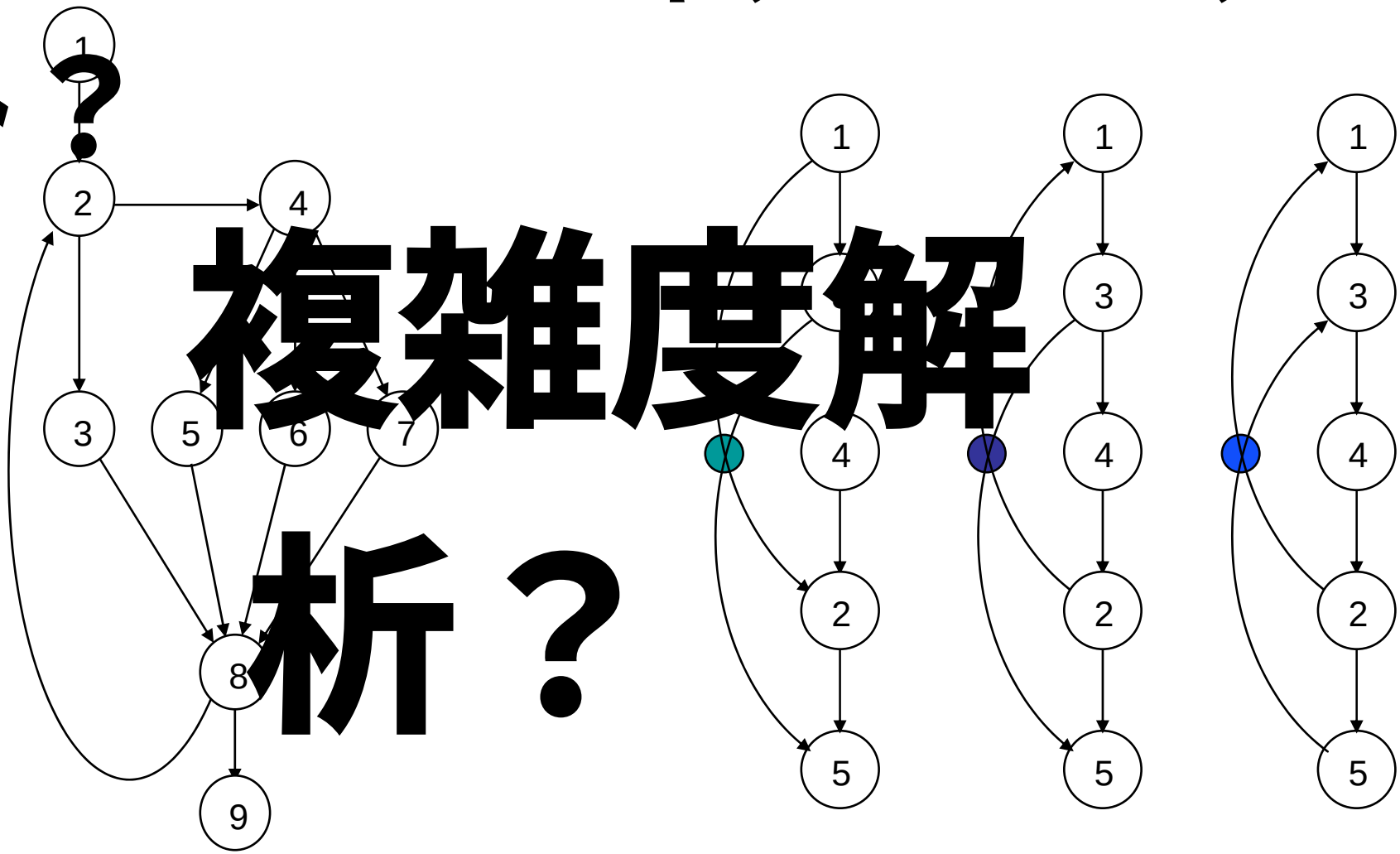
複雑度解析

サイクロマティック？



複雑度解析？

サイクロマティック？ ノット？



複雑度解析？

サイクロマティック？ノット？複雑度解析？

•Executable ref. Lines

•Executable reformatted Lines

•Essential Knots

•Total Comments

•Number of Basic Blocks

•Essential Cyclomatic Complexity

•Comments in Headers

•Total Operands

•Knots

•Comments in Declarations

•Number of Loops

•Cyclomatic Complexity

•Comments in Executable Code

•Procedure Exit Points

•Vocabulary

•Blank Lines

•Number of Procedures

•Total Comments/Exe. Lines

•Total LCSAJs

•Unreachable LCSAJs

•Declaration Comments/Exe. Lines

•Maximum LCSAJ Density

•Maximum LCSAJ Density

•Code Comments/Exe. Lines

•Unreachable Lines

•Unreachable Lines

•Average Length of Basic Blocks

•Unreachable Branches

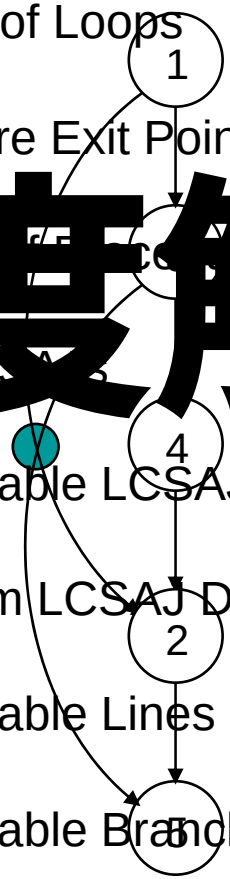
•Unreachable Branches

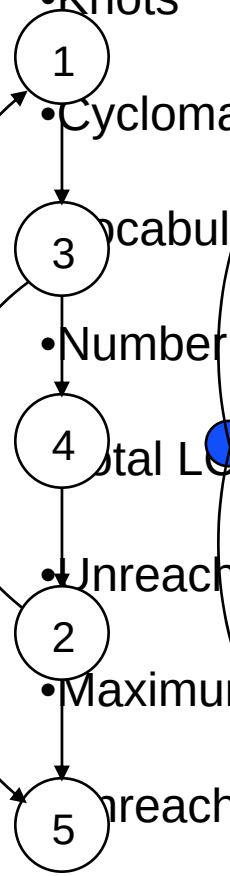
•Unique Operands

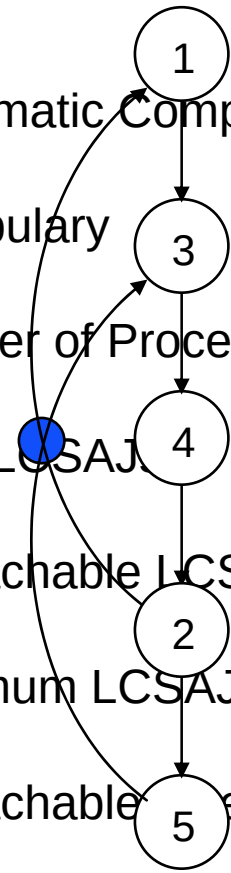
•File Fan in

•Total LCSAJs

•Fan Out







サイクロマティック？ノット？複雑度解析？キビアット図？

•Executable ref. Lines

•Total Comments

•Comments in Headers

•Comments in Declarations

•Comments in Executable Code

•Blank Lines

•Total Comments

•Declaration Comment Lines

•Code Comments

•Average Length

•Unique Operands

•Total LCSAJs

•Executable reformatted Lines

•Number of Basic Blocks

•Total Operands

•Number of Loops

•Procedure Exit Points

•Total LCSAJs

•Fan Out

•Essential Knots

•Essential Cyclomatic Complexity

•Knots

•Cyclomatic Complexity

•Vocabulary

•Number of Procedures

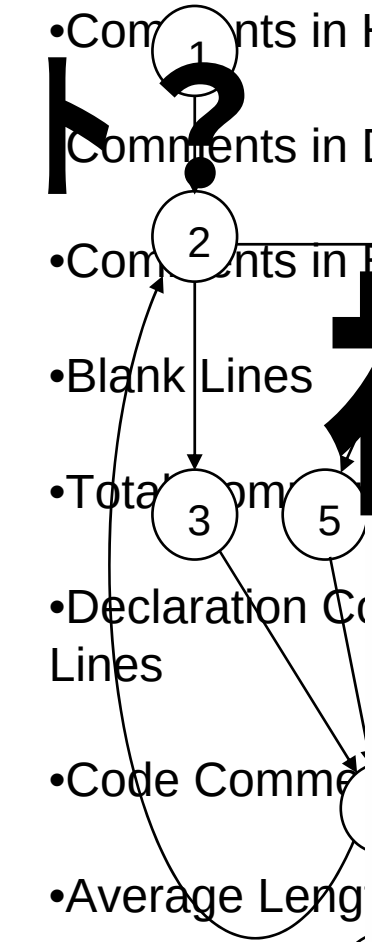
•Total LCSAJs

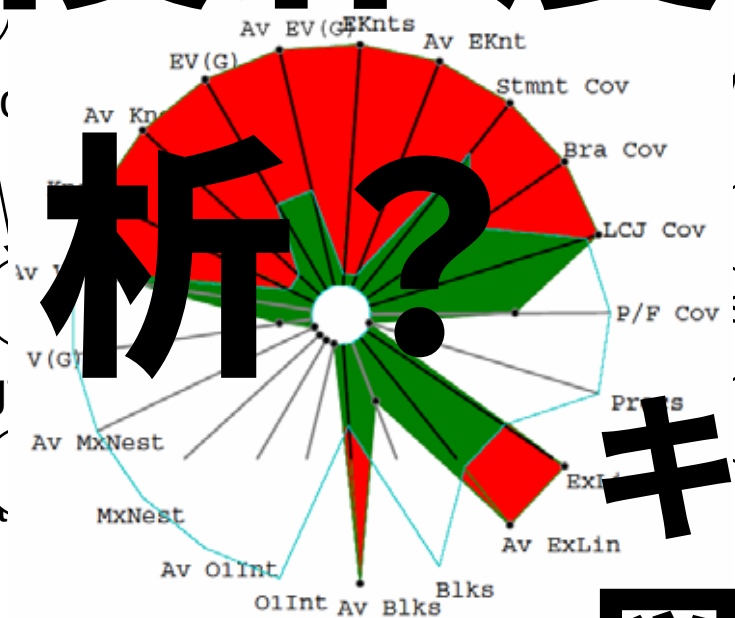
•Unreachable LCSAJs

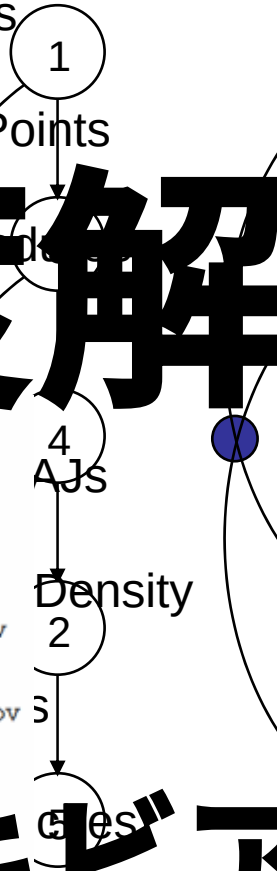
•Maximum LCSAJ Density

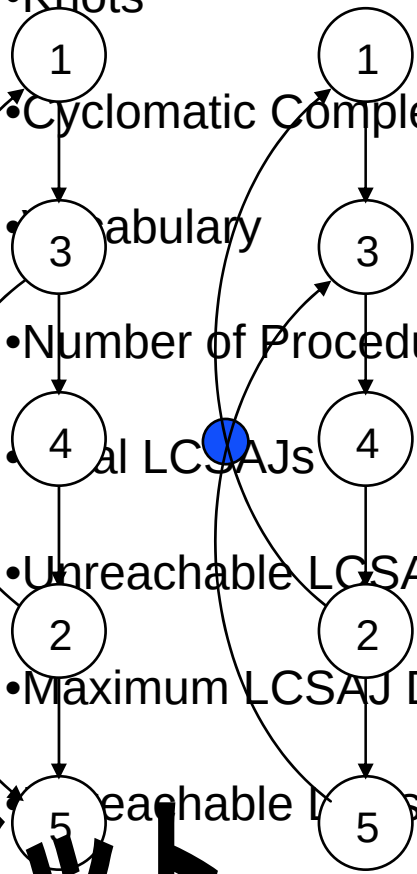
•Unreachable Blocks

•Unreachable Branches









でも、 

の

静的テスト

であれば、

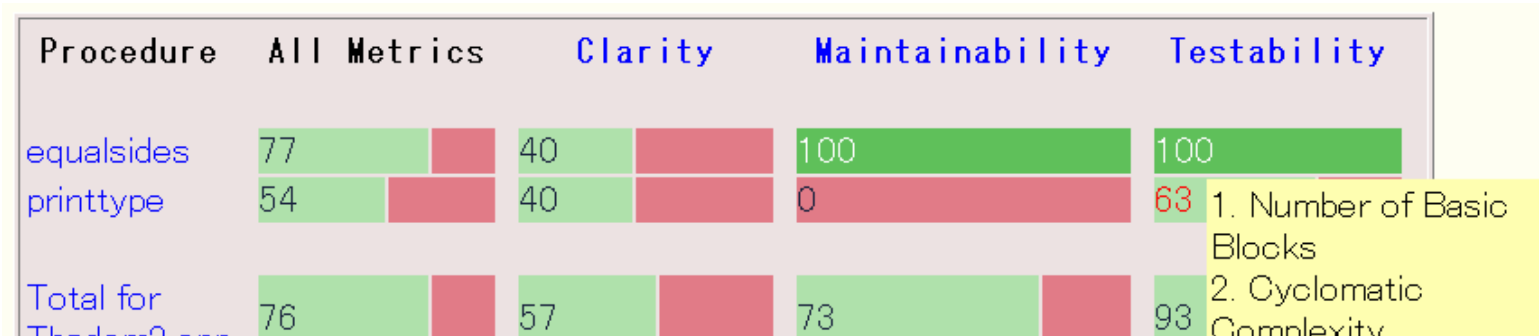
解析済の領域は、

抑止できる。

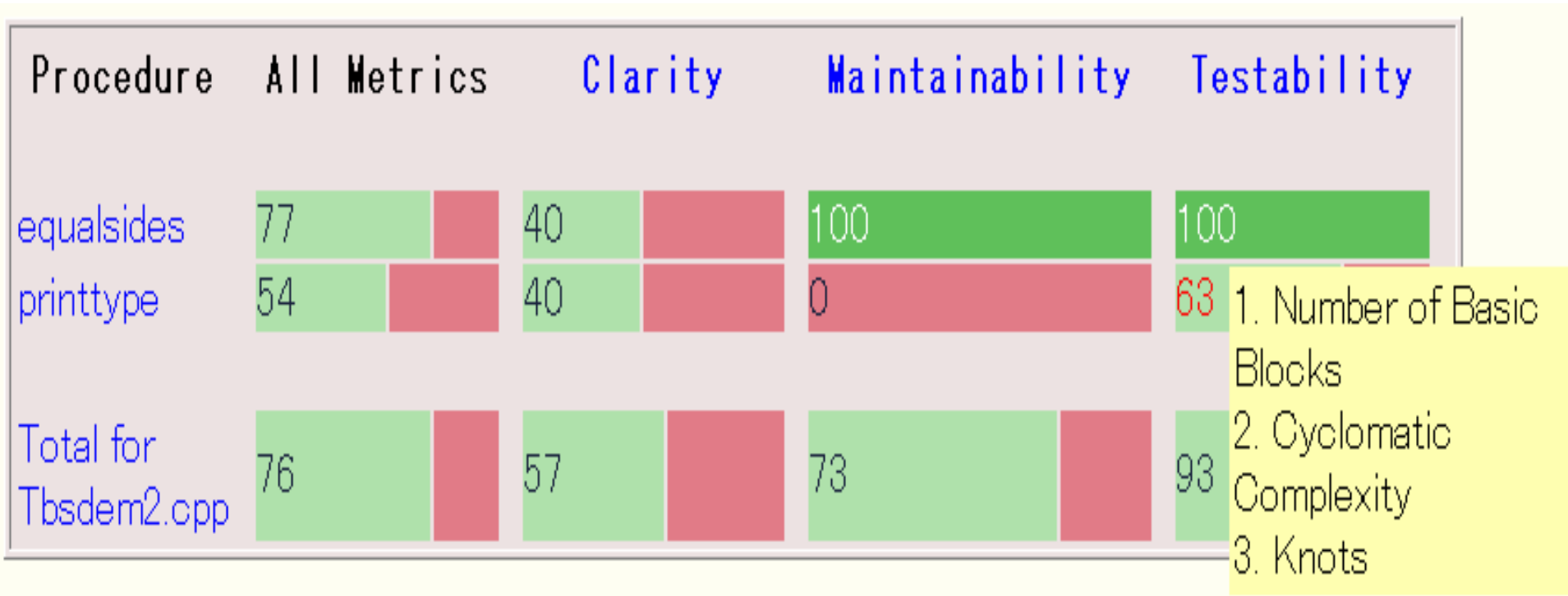
/*LDRA_INSPECTED */

複雑度解析も、

可読性、メンテナンス容易性、テスト容易性に



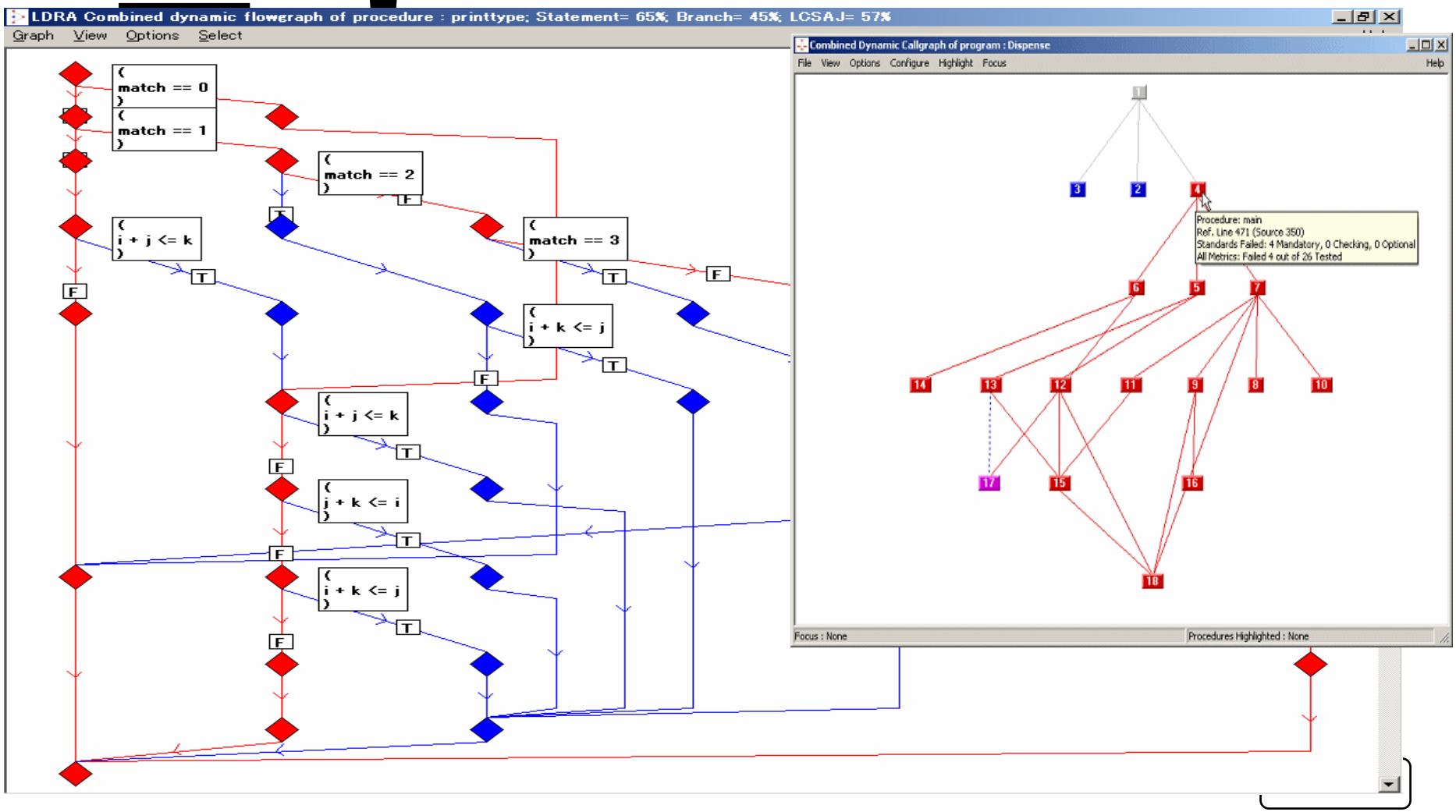
おまとめ表示。


Procedure
All Metrics
Clarity
Maintainability
Testability
equalsides
77
40
100
100
printtype
54
40
0
63
Total for Tbsdem2.cpp
76
57
73
93
1. Number of Basic Blocks
2. Cyclomatic Complexity
3. Knots

しかも、

単体テスト
テストカバレッジ
が、

視覚的に融合！


LDRA Combined dynamic flowgraph of procedure : printtype; Statement= 65%; Branch= 45%; LCSAJ= 57%
Graph View Options Select
match == 0
match == 1
match == 2
match == 3
i + j <= k
i + k <= j
i + j <= k
j + k <= i
i + k <= j
Combined Dynamic Callgraph of program : Dispense
File View Options Configure Highlight Focus Help
Procedure: main
Ref. Line 471 (Source 350)
Standards Failed: 4 Mandatory, 0 Checking, 0 Optional
All Metrics: Failed 4 out of 26 Tested
Focus : None
Procedures Highlighted : None

検証作業、

リグレション、

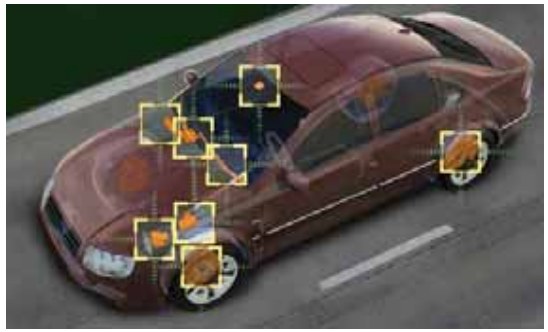
などを効率化

・ソースコードベース

・**モデル検証**

・自動コード生成

（Domain-Specific Modeling）

TVEC

We Cover All Boundaries

モデル検証

モデル検証の課題は、、、

形式的仕様記述言語を使って

要求仕様を？？？

```
VARIABLE Crew_Interface_Throttles_term_GA_Pressed__VAR ISA Boolean ;


LOGIC STRUCTURE Autopilot_Modes_Disengaged_Submode_FROM_CLEARED_NoChange__LS(_Autopilot_Modes_Disengaged_Submode:A
  ATTRIBUTE ModelLoc="ttm://FGS?name=Autopilot_Modes_Disengaged_Submode";
  ATTRIBUTE ModelLocID="No-change for mode CLEARED";
  ATTRIBUTE ReqID="No-change for mode CLEARED";
  CONSTRAINT
              ((_Autopilot_Modes_Disengaged_Submode = CLEARED::Autopilot_Modes_Disengaged_Submode__Type)
                  AND
              NOT:(_Autopilot_Modes_Engage = DISENGAGED::Autopilot_Modes_Engage__Type))
                  AND
              (((_Autopilot_Modes_Disengaged_Submode = CLEARED::Autopilot_Modes_Disengaged_Submode__Type)
                  AND
              (_Autopilot_Modes_Engage = Autopilot_Modes_Engage__VAR))
                  AND
              (_Crew_Interface_Throttles_term_GA_Pressed = Crew_Interface_Throttles_term_GA_Pressed__VAR));

LOGIC STRUCTURE Autopilot_Modes_Disengaged_Submode_FROM_CLEARED_TO_Normal__D1__LS(_Autopilot_Modes_Disengaged_Subm
  ATTRIBUTE ModelLoc="ttm://FGS?name=Autopilot_Modes_Disengaged_Submode&row=3&col=2";
  CONSTRAINT
              (_Autopilot_Modes_Disengaged_Submode = CLEARED::Autopilot_Modes_Disengaged_Submode__Type)
                  AND
              ((NOT:(_Autopilot_Modes_Engage = DISENGAGED::Autopilot_Modes_Engage__Type)
                  AND
              (Autopilot_Modes_Engage__VAR = DISENGAGED::Autopilot_Modes_Engage__Type))
                  AND
              NOT:(_Crew_Interface_Throttles_term_GA_Pressed = FALSE));

LOGIC STRUCTURE Autopilot_Modes_Disengaged_Submode_FROM_CLEARED_TO_Normal__D2__LS(_Autopilot_Modes_Disengaged_Subm
  ATTRIBUTE ModelLoc="ttm://FGS?name=Autopilot_Modes_Disengaged_Submode&row=3&col=2";
  CONSTRAINT
```

モデルの検証で終わり？？

で、

実装の検証はどーなるの？

というのが、一般的なモデル検証、フォーマルメソッドの課題

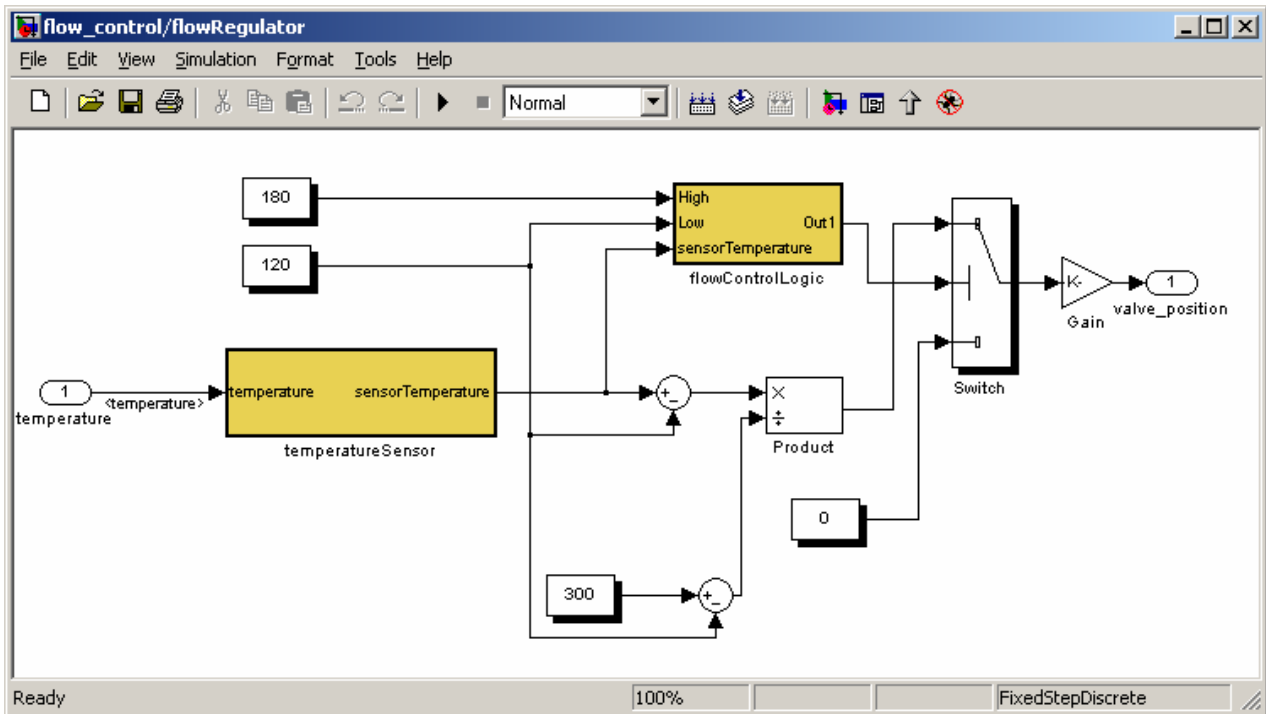
We Cover All Boundaries

モデル検証は、

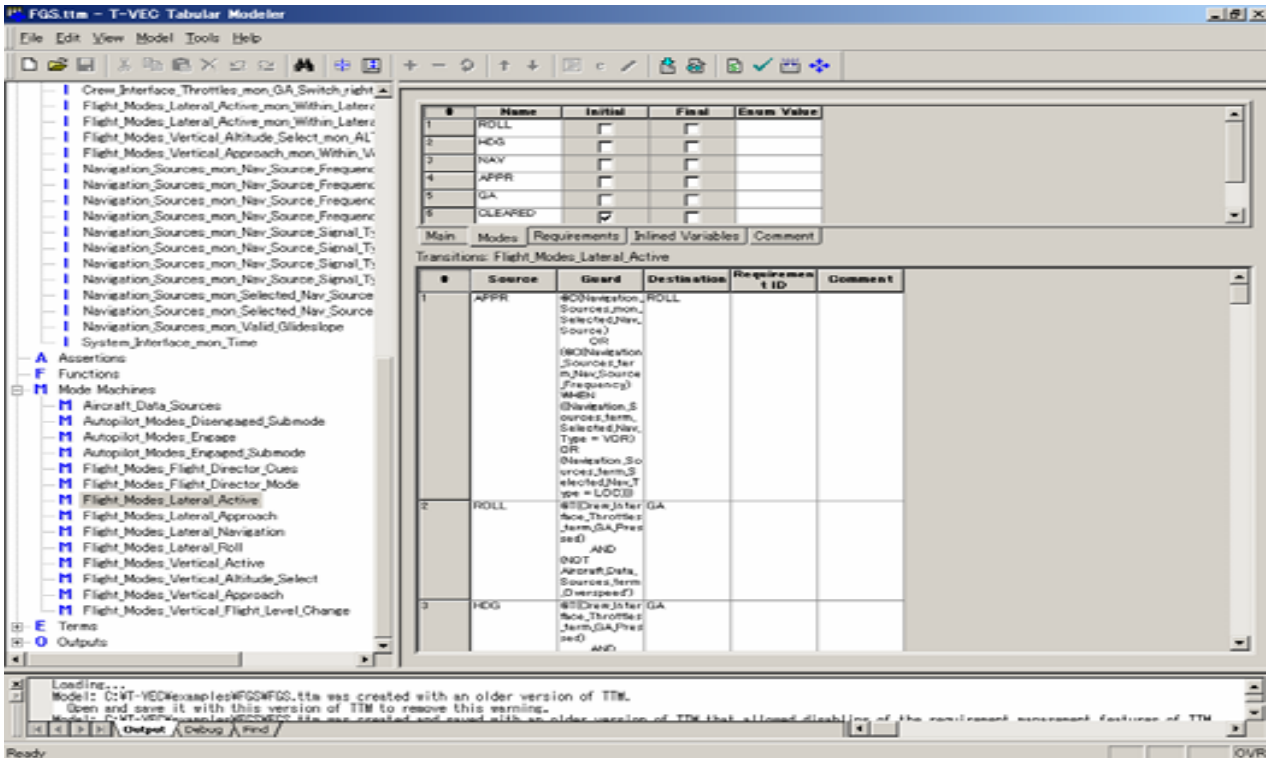
形式的仕様記述言語を

Simulink

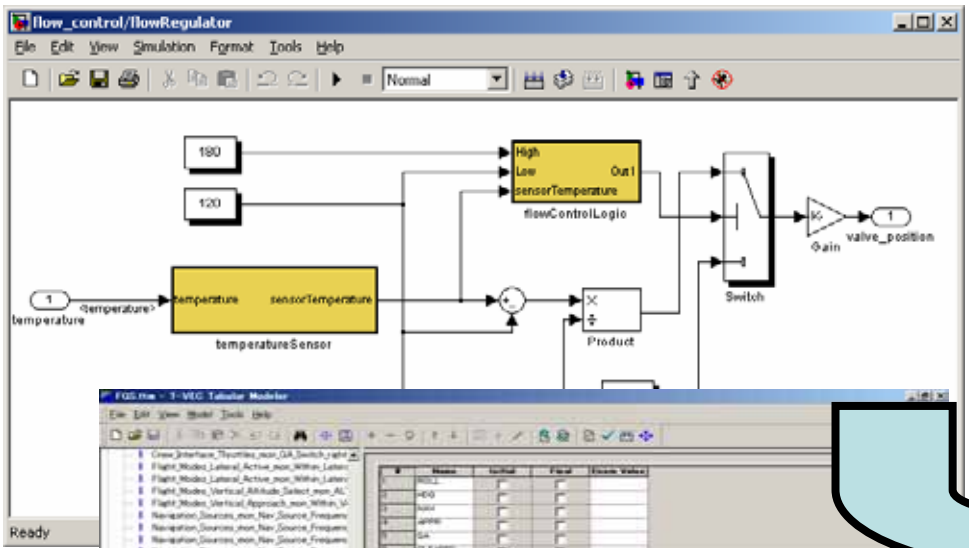
モデルや

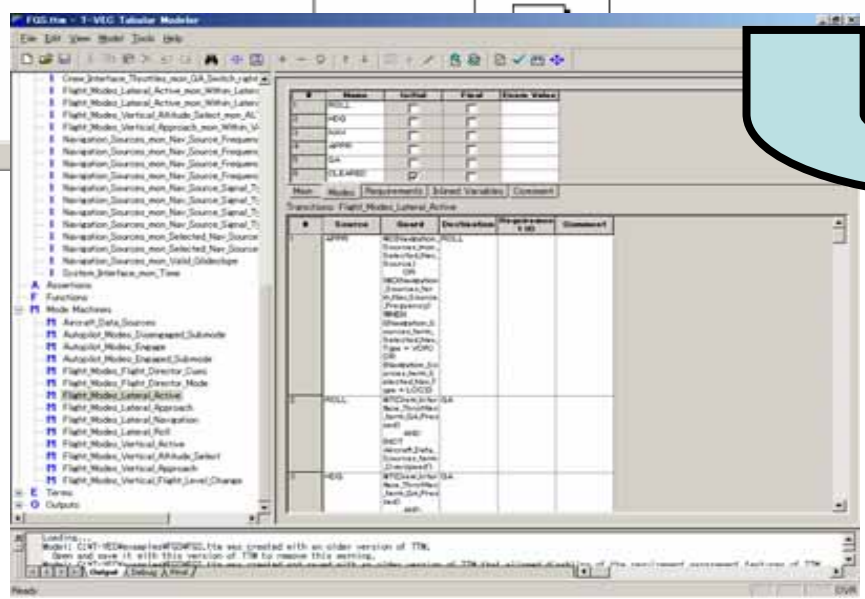


TTM 要求仕様の表モデルを



変換して生成。





```
VARIABLE Crew_Interface_Throttles_term_GA_Pressed__VAR ISA Boolean ;


LOGIC STRUCTURE Autopilot_Modes_Disengaged_Submode_FROM_CLEARED_NoChange__LS(_Autopilot_Modes_Disengaged_Submode:A
  ATTRIBUTE ModelLoc="ttm://FGS?name=Autopilot_Modes_Disengaged_Submode";
  ATTRIBUTE ModelLocID="No-change for mode CLEARED";
  ATTRIBUTE ReqID="No-change for mode CLEARED";
  CONSTRAINT
            ((_Autopilot_Modes_Disengaged_Submode = CLEARED::Autopilot_Modes_Disengaged_Submode__Type)
               AND
            NOT:(_Autopilot_Modes_Engage = DISENGAGED::Autopilot_Modes_Engage__Type))
               AND
            (((_Autopilot_Modes_Disengaged_Submode = CLEARED::Autopilot_Modes_Disengaged_Submode__Type)
               AND
            (_Autopilot_Modes_Engage = Autopilot_Modes_Engage__VAR))
               AND
            (_Crew_Interface_Throttles_term_GA_Pressed = Crew_Interface_Throttles_term_GA_Pressed__VAR));

LOGIC STRUCTURE Autopilot_Modes_Disengaged_Submode_FROM_CLEARED_TO_Normal__D1__LS(_Autopilot_Modes_Disengaged_Subm
  ATTRIBUTE ModelLoc="ttm://FGS?name=Autopilot_Modes_Disengaged_Submode&row=8&col=2";
  CONSTRAINT
            (_Autopilot_Modes_Disengaged_Submode = CLEARED::Autopilot_Modes_Disengaged_Submode__Type)
               AND
            ((NOT:(_Autopilot_Modes_Engage = DISENGAGED::Autopilot_Modes_Engage__Type)
               AND
            (Autopilot_Modes_Engage__VAR = DISENGAGED::Autopilot_Modes_Engage__Type))
               AND
            NOT:(_Crew_Interface_Throttles_term_GA_Pressed = FALSE));

LOGIC STRUCTURE Autopilot_Modes_Disengaged_Submode_FROM_CLEARED_TO_Normal__D2__LS(_Autopilot_Modes_Disengaged_Subm
  ATTRIBUTE ModelLoc="ttm://FGS?name=Autopilot_Modes_Disengaged_Submode&row=8&col=2";
  CONSTRAINT
```

楽です。

直接書くより。

```
VARIABLE Crew_Interface_Throttles_term_GA_Pressed__VAR ISA Boolean ;


LOGIC STRUCTURE Autopilot_Modes_Disengaged_Submode_FROM_CLEARED_NoChange__LS(_Autopilot_Modes_Disengaged_Submode:A
  ATTRIBUTE ModelLoc="ttm://FGS?name=Autopilot_Modes_Disengaged_Submode";
  ATTRIBUTE ModelLocID="No-change for mode CLEARED";
  ATTRIBUTE ReqID="No-change for mode CLEARED";
  CONSTRAINT
              ((_Autopilot_Modes_Disengaged_Submode = CLEARED::Autopilot_Modes_Disengaged_Submode__Type)
                  AND
              NOT:(_Autopilot_Modes_Engage = DISENGAGED::Autopilot_Modes_Engage__Type))
                  AND
              (((_Autopilot_Modes_Disengaged_Submode = CLEARED::Autopilot_Modes_Disengaged_Submode__Type)
                  AND
              (_Autopilot_Modes_Engage = Autopilot_Modes_Engage__VAR))
                  AND
              (_Crew_Interface_Throttles_term_GA_Pressed = Crew_Interface_Throttles_term_GA_Pressed__VAR));

LOGIC STRUCTURE Autopilot_Modes_Disengaged_Submode_FROM_CLEARED_TO_Normal__D1__LS(_Autopilot_Modes_Disengaged_Subm
  ATTRIBUTE ModelLoc="ttm://FGS?name=Autopilot_Modes_Disengaged_Submode&row=3&col=2";
  CONSTRAINT
              (_Autopilot_Modes_Disengaged_Submode = CLEARED::Autopilot_Modes_Disengaged_Submode__Type)
                  AND
              ((NOT:(_Autopilot_Modes_Engage = DISENGAGED::Autopilot_Modes_Engage__Type)
                  AND
              (Autopilot_Modes_Engage__VAR = DISENGAGED::Autopilot_Modes_Engage__Type))
                  AND
              NOT:(_Crew_Interface_Throttles_term_GA_Pressed = FALSE));

LOGIC STRUCTURE Autopilot_Modes_Disengaged_Submode_FROM_CLEARED_TO_Normal__D2__LS(_Autopilot_Modes_Disengaged_Subm
  ATTRIBUTE ModelLoc="ttm://FGS?name=Autopilot_Modes_Disengaged_Submode&row=3&col=2";
  CONSTRAINT
```

しかも、

モデルの検証で終わらない！

モデル検証結果からテストベクタを生成！

Test Vectors (Compressed) For Subsystem : temperatureSensor

Test #	Vector #'s	__temperatureSensorData ._output	__temperatureSensorData .__local .IState_Unit_Delay	temperature_iPort	__local .IState_Unit_Delay
1	1, 9	-1.0e+002	-2.457e+002	-2.73e+002	0.0e+000
2	2, 6	3.0e+002	5.4e+002	6.0e+002	0.0e+000
3	3, 11	-1.0e+002	-1.00000002457e+010	-2.73e+002	-1.0e+011
4	4, 8	3.0e+002	1.000000054e+010	6.0e+002	1.0e+011
5	5	3.0e+002	3.00000000000055e+002	3.33333333333394e+002	0.0e+000
6	7	3.0e+002	3.0000000000002e+002	-2.73e+002	5.4570000000002e+003
7	10	-1.0e+002	-1.0000000000012e+002	-1.11111111111244e+002	0.0e+000
8	12	-1.0e+002	-1.00002e+002	6.0e+002	-6.40002e+003
9	13	-1.0e+002	-1.0e+002	-1.11111111111111e+002	0.0e+000
10	14	3.0e+002	3.0e+002	3.33333333333333e+002	0.0e+000
11	15	-1.0e+002	-1.0e+002	-2.73e+002	1.457e+003
12	16	3.0e+002	3.0e+002	6.0e+002	-2.4e+003

Test #1: Vector(s) : 1, 9

```
DCP # 1 : temperatureSensor, temperatureSensor_FR__1, cv_temperatureSensor_RP__1
temperatureSensor_RP__0, RP0, RP1_1
Constrained Input CV Mode     : LOW_BOUND
UnConstrained Input CV Mode   : OPPOSITE
Vector Generation Mode        : SINGLE
Input Value Assignment Order  : SPEC_SEQUENTIAL_INPUT
MCDC Support Level            : DO178B_MCDC
```

期待値と、　入力値

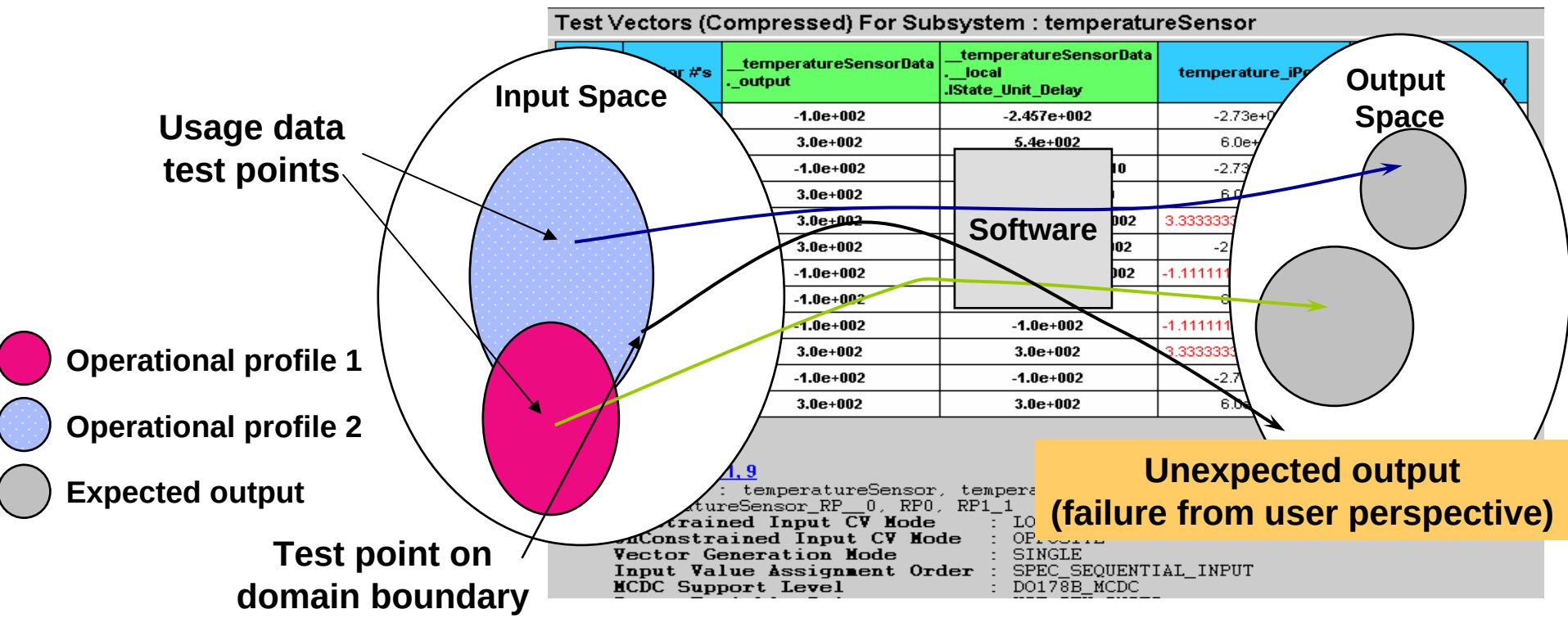
Test Vectors (Compressed) For Subsystem : temperatureSensor

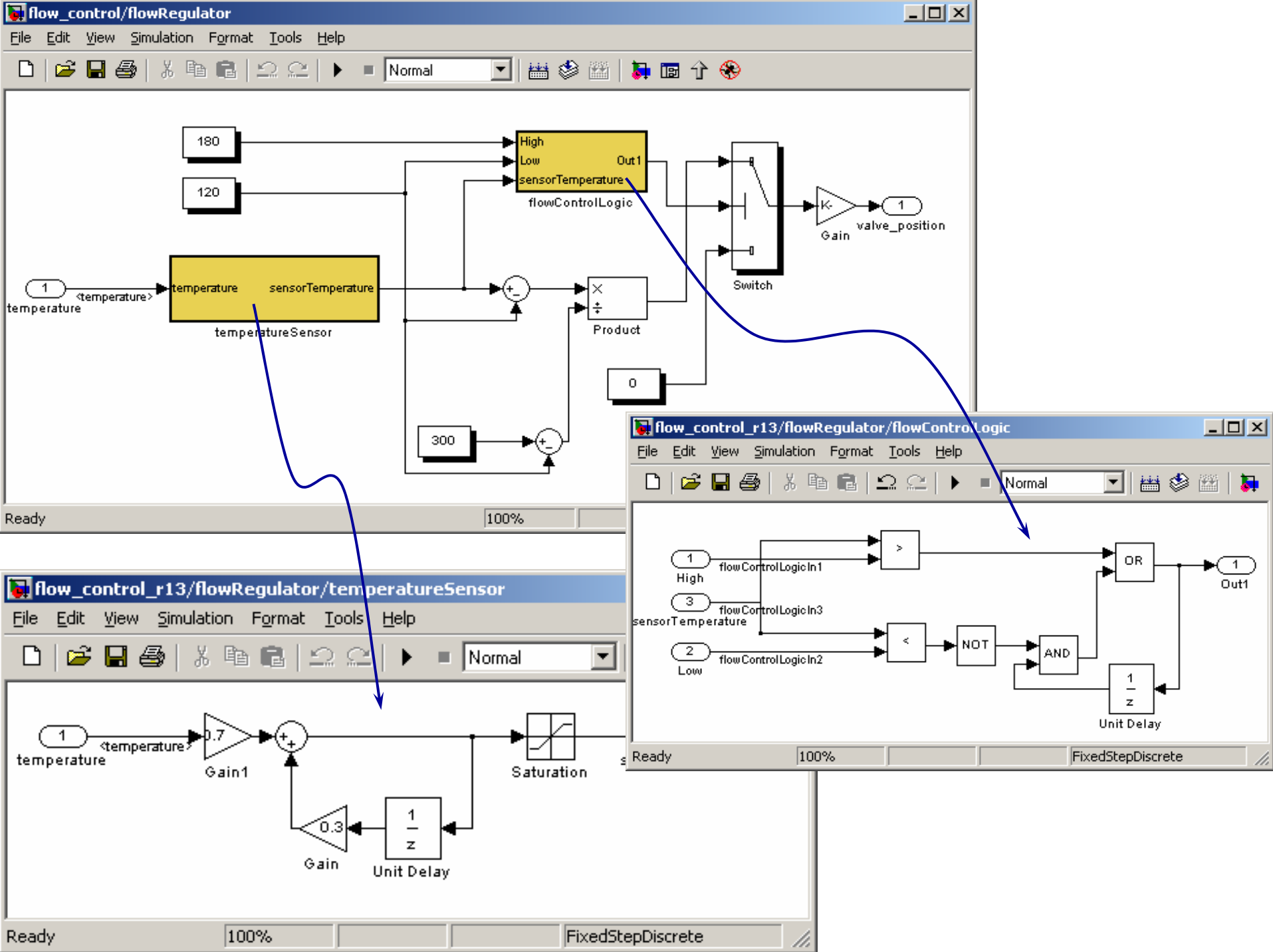
Test #	Vector #'s	__temperatureSensorData ._output	__temperatureSensorData .__local .IState_Unit_Delay	temperature_iPort	__local .IState_Unit_Delay
1	1, 9	-1.0e+002	-2.457e+002	-2.73e+002	0.0e+000
2	2, 6	3.0e+002	5.4e+002	6.0e+002	0.0e+000
3	3, 11	-1.0e+002	-1.00000002457e+010	-2.73e+002	-1.0e+011
4	4, 8	3.0e+002	1.000000054e+010	6.0e+002	1.0e+011
5	5	3.0e+002	3.00000000000055e+002	3.33333333333394e+002	0.0e+000
6	7	3.0e+002	3.0000000000002e+002	-2.73e+002	5.4570000000002e+003
7	10	-1.0e+002	-1.0000000000012e+002	-1.11111111111244e+002	0.0e+000
8	12	-1.0e+002	-1.00002e+002	6.0e+002	-6.40002e+003
9	13	-1.0e+002	-1.0e+002	-1.11111111111111e+002	0.0e+000
10	14	3.0e+002	3.0e+002	3.33333333333333e+002	0.0e+000
11	15	-1.0e+002	-1.0e+002	-2.73e+002	1.457e+003
12	16	3.0e+002	3.0e+002	6.0e+002	-2.4e+003

さらに凄いことに

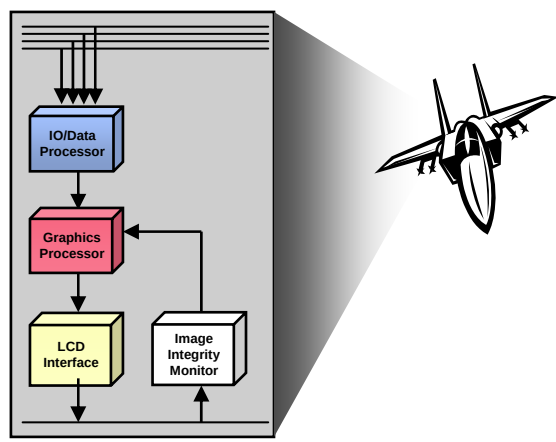
テストベクタは、

サブドメインの境界値を攻める。

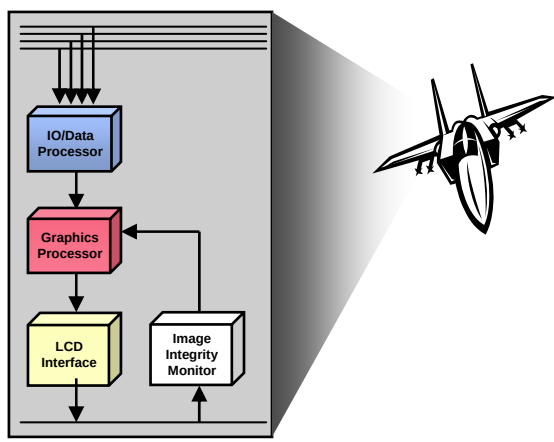



flow_control/flowRegulator
File Edit View Simulation Format Tools Help
Normal
180
120
High
Low
sensorTemperature
Out1
flowControlLogic
K-
Gain
1
valve_position
Switch
1
temperature
<temperature>
temperature
sensorTemperature
temperatureSensor
Product
0
300
Ready
100%
flow_control_r13/flowRegulator/flowControlLogic
File Edit View Simulation Format Tools Help
Normal
1
High
flowControlLogicIn1
3
sensorTemperature
flowControlLogicIn3
2
Low
flowControlLogicIn2
>
<
NOT
AND
OR
1
Out1
1
z
Unit Delay
Ready
100%
FixedStepDiscrete
flow_control_r13/flowRegulator/temperatureSensor
File Edit View Simulation Format Tools Help
Normal
1
temperature
<temperature>
0.7
Gain1
Saturation
0.3
Gain
1
z
Unit Delay
Ready
100%
FixedStepDiscrete

そして、

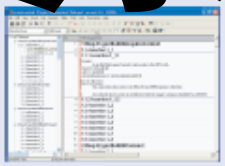

IO/Data Processor
Graphics Processor
LCD Interface
Image Integrity Monitor

あらゆる環境でモデルとの一致性検証に用いられる



Full Lifecycle Model-Based Verification

要求モデル



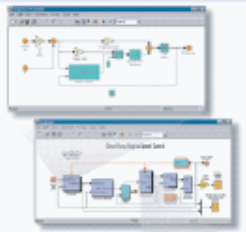
DOORS®
Requirements
Management

T-VEC®
Tabular Modeler

Requirements
Errors

System Test
Vectors

System Test
Scripts/Drivers



Simulink® and
Stateflow® Design

デザインモデル

T-VEC® Tester
for Simulink®
and Stateflow®

Design
Errors

Unit Test
Vectors

Unit Test
Drivers
(C and Simulink)

Implementation through
Real-Time Workshop®
and Stateflow® Coder

Telelogic
Partner

SIMULINK®
Enabled
MathWorks Partner

英国防省にも採用を推奨され、

DO178C のモデルベース開発ガイドライン

にも関わっています。

NIST の SmartCard

モデル検証など

NIST
National Institute of Standards and Technology
Technology Administration
U.S. Department of Commerce

Interagency Report 6887 - 2003 Edition

Government Smart Card Interoperability Specification

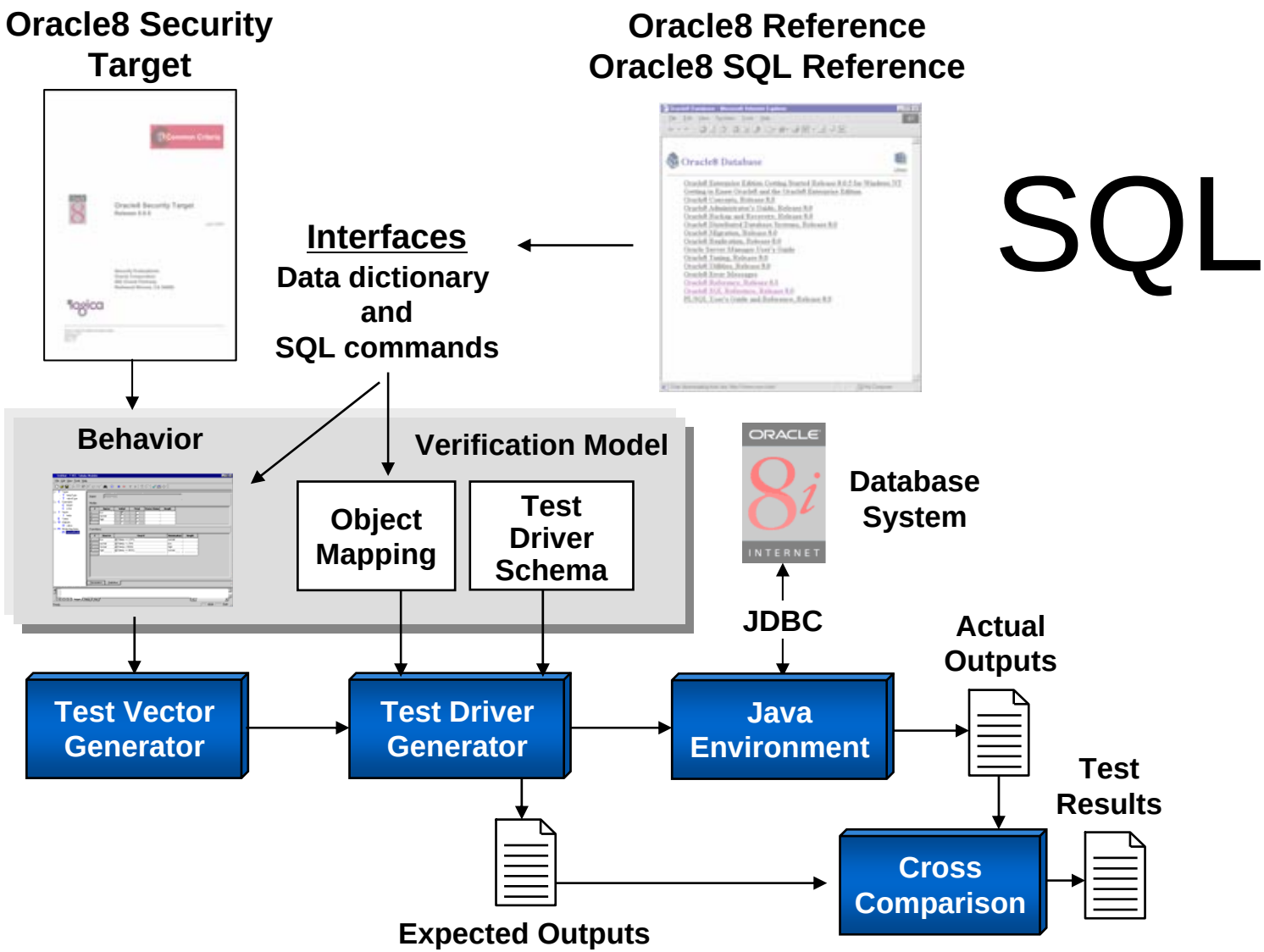
Version 2.1

Teresa Schwarzhoff
Jim Dray
John Wack
Eric Dalci
Alan Goldfine
Michaela Iorga

米国政府機関スマートカードの相互運用に関する仕様書

July 16, 2003

データベースの安全性要求検証
米) 国立標準技術研究所 (NIST)

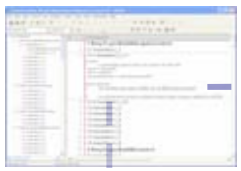


Example 2

実践的に産業界で採用されるモデル検証

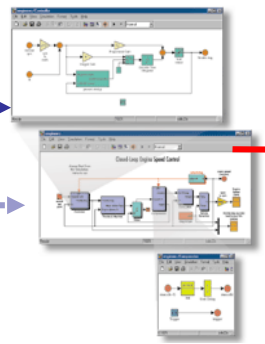
要求仕様～ Simulink モデル～テストのトレーサビリティー

要求仕様書や、
要件管理ツール
(DOORS など)

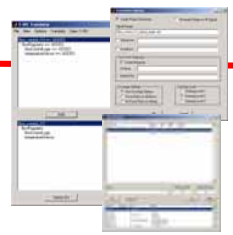


T-VEC (TTM)
要求仕様を
表でモデル表記

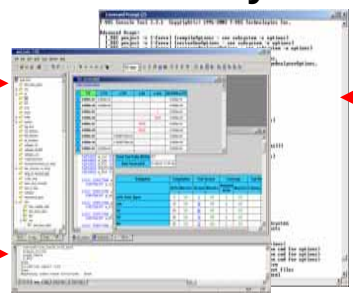
Simulink
Stateflow



Simulink
Tester GUI



T-VEC Test Vector
Generation System



- 要求仕様を表でモデル表記
- 曖昧な要素を形式的モデルに変換

- デザインモデル
- シミュレーション
- 自動コード生成

- モデル検証
- テストベクタ生成
- モデルテストカバレッジ
- テストドライバー生成
- テスト結果解析

- コードカバレッジ (MCDCなど)
- 静的解析 (MISRA など)

Requirements/Design Capture

Captured Model Translation

ロッキードマーチン社によるJSF開発事例

- ソースコードベース
- モデル検証
- **自動コード生成**（Domain-Specific Modeling）

Domain-Specific Modeling
for Full Code Generation

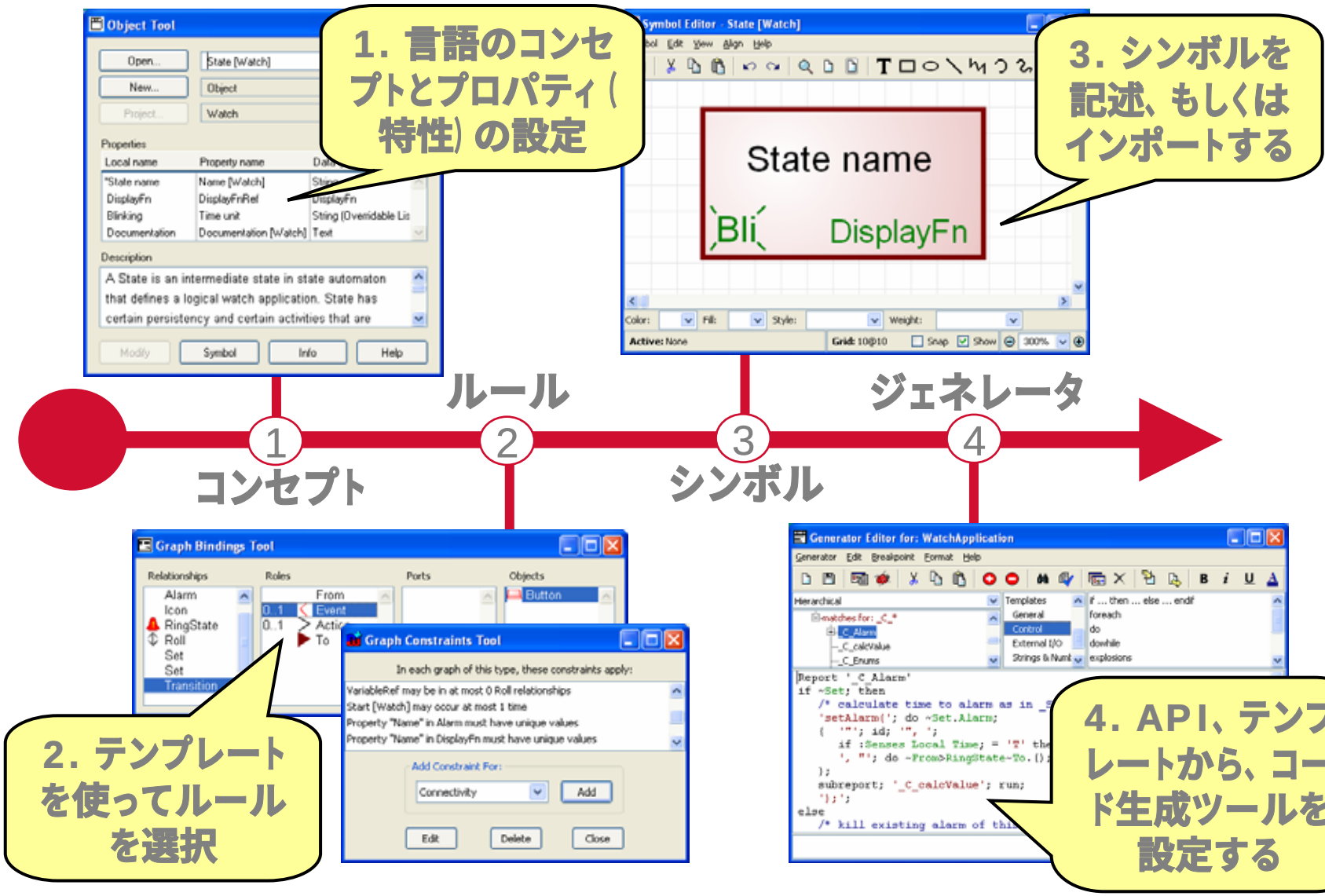
自動コード生成

MetaCase

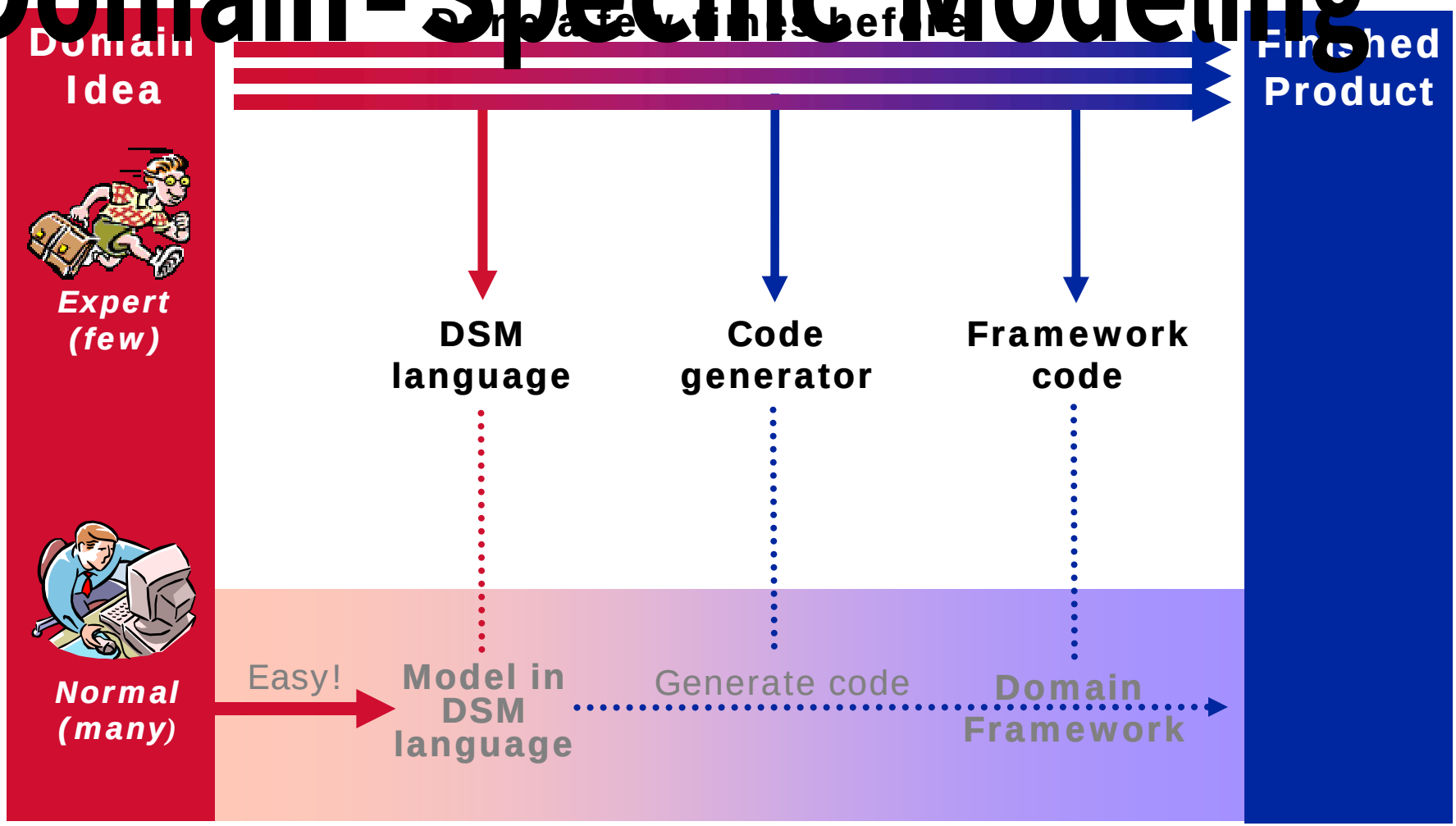
Domain-Specific Modeling
for Full Code Generation

は、

DSMを作るためのツール



DSMは、
Domain-Specific Modeling



企業、製品のドメインごとに、

- Domain-Specific にモデリングし、
- Domain-Specific なコードを生成させる

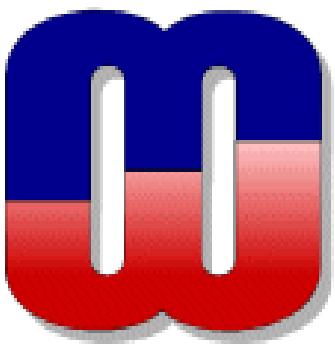
企業、製品のドメインごとに、

- Domain-Specific にモデリングし、
- Domain-Specific なコードを生成させる

企業、製品のドメインごとに、

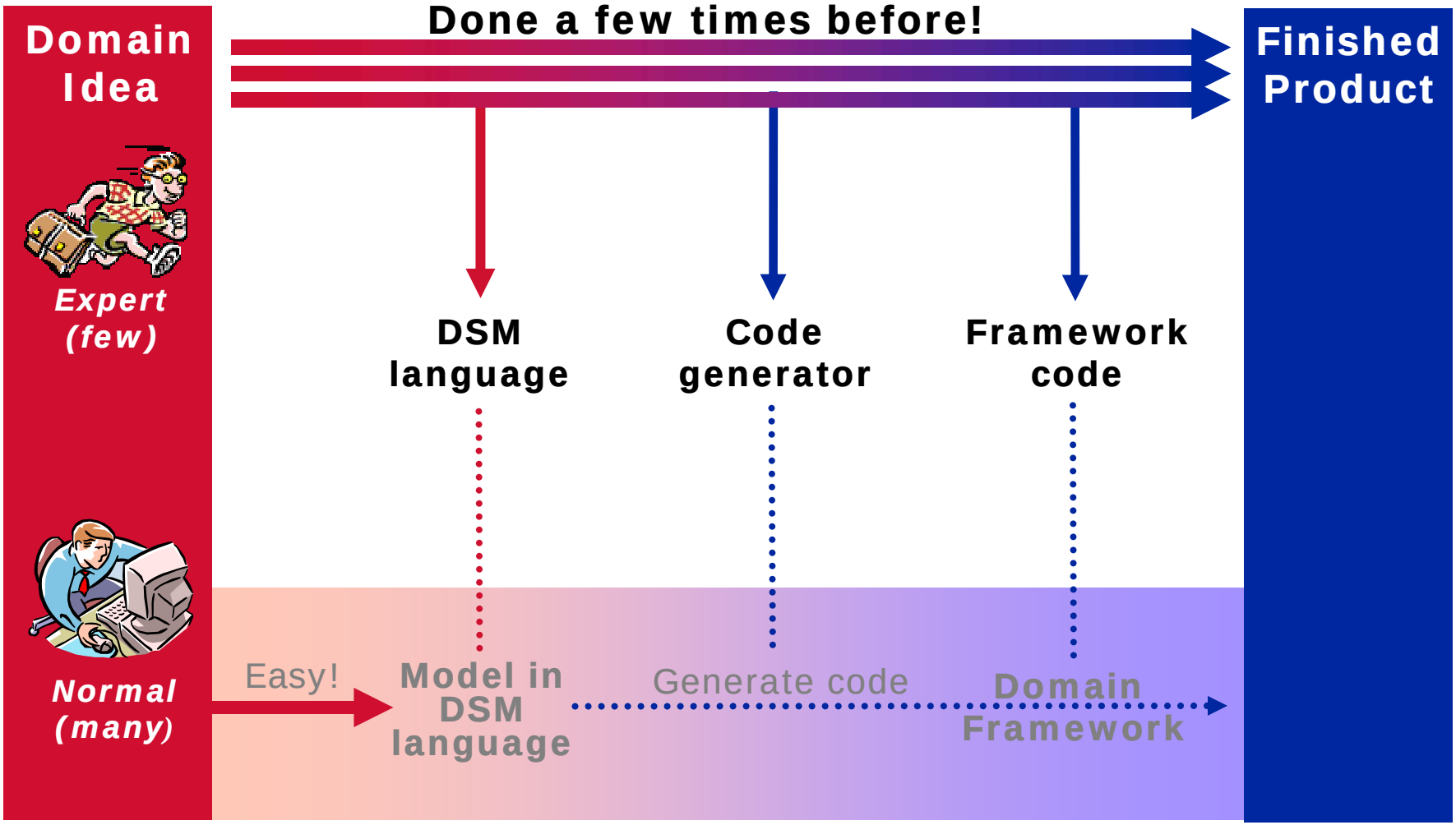
- Domain-Specific にモデリングし、
- Domain-Specific なコードを生成させる

そんなフレームワークを作る為のツールが、

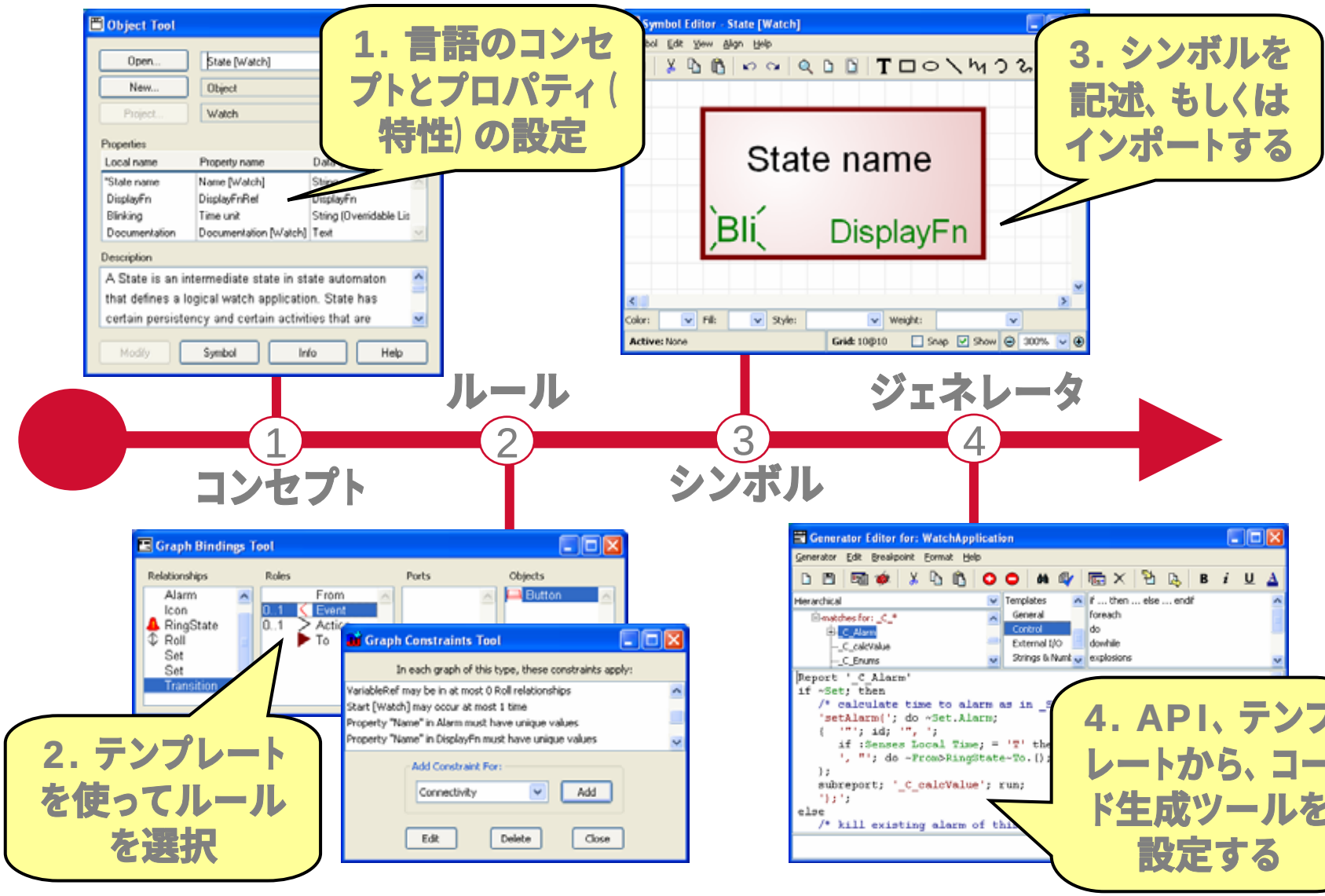


MetaEdit+ です。

ドメイン精通者がDSM作成
皆は、DSMを使って開発

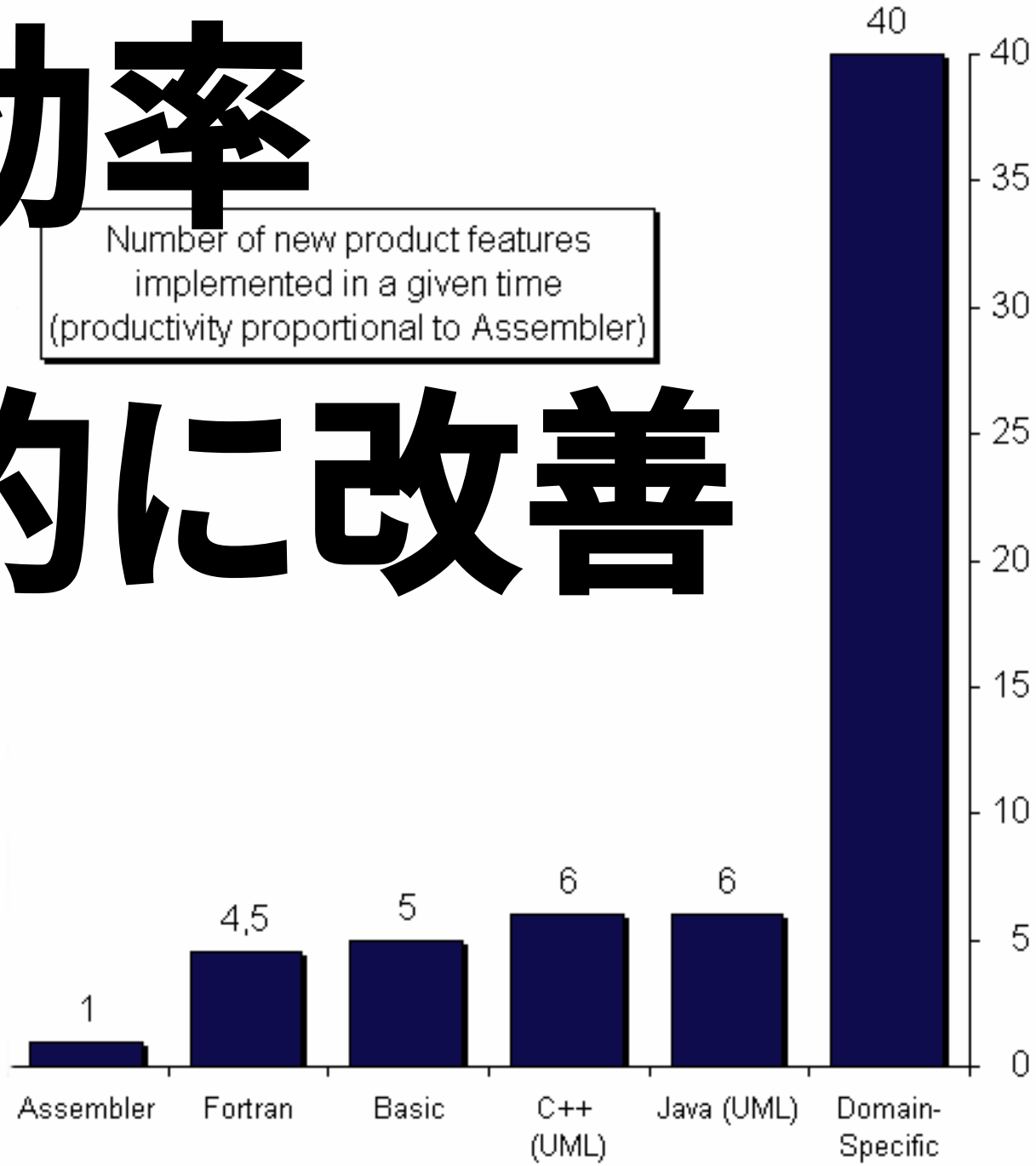


抽象度をモデルに高めて、



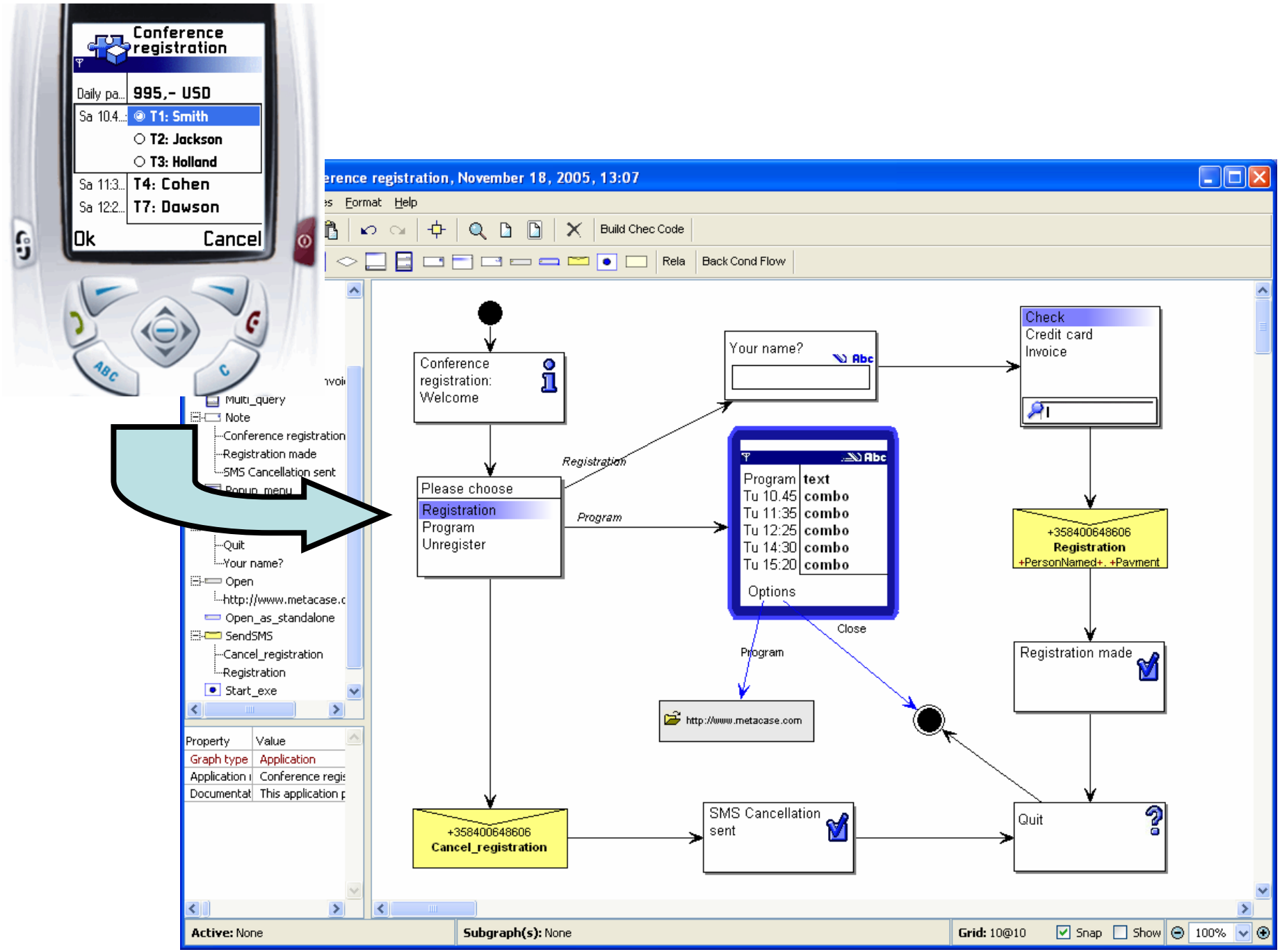
開発効率

飛躍的に改善

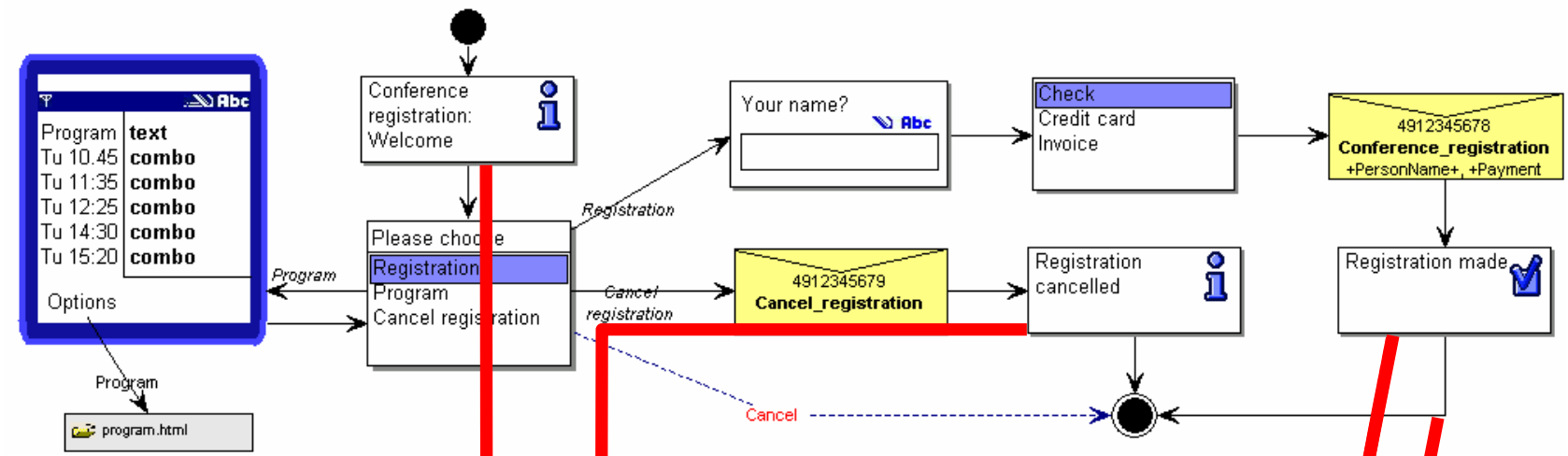

Number of new product features
implemented in a given time
(productivity proportional to Assembler)
40
40
35
30
25
20
15
10
5
0
1
4,5
5
6
6
Assembler
Fortran
Basic
C++
(UML)
Java (UML)
Domain-
Specific

Case1: スマートフォン

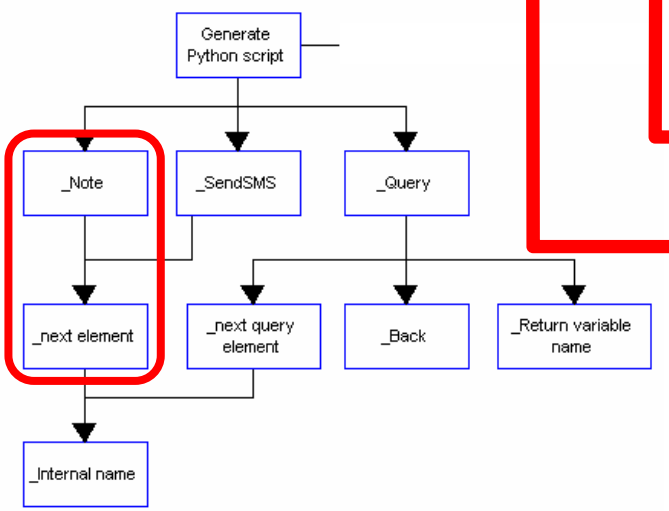
Symbian / Series 60


Conference registration
Daily pa...
995,- USD
Sa 10.4...
T1: Smith
T2: Jackson
T3: Holland
Sa 11:3...
T4: Cohen
Sa 12:2...
T7: Dawson
Ok
Cancel
ABC
erence registration, November 18, 2005, 13:07
Format
Help
Build Chec Code
Rela
Back Cond Flow
Multi_query
Note
Conference registration
Registration made
SMS Cancellation sent
Quit
Your name?
Open
http://www.metacase.c
Open_as_standalone
SendSMS
Cancel_registration
Registration
Start_exe
Property
Value
Graph type
Application
Application
Conference regis
Documentat
This application
Conference registration: Welcome
Please choose
Registration
Program
Unregister
Registration
Program
Your name?
Abc
Program
text
Tu 10.45
combo
Tu 11:35
combo
Tu 12:25
combo
Tu 14:30
combo
Tu 15:20
combo
Options
Close
Program
http://www.metacase.com
Check
Credit card
Invoice
+358400648606
Registration
+PersonNamed+, +Payment
Registration made
Quit
+358400648606
Cancel_registration
SMS Cancellation sent
Active: None
Subgraph(s): None
Grid: 10@10
Snap
Show
100%

Function calls – Series 60



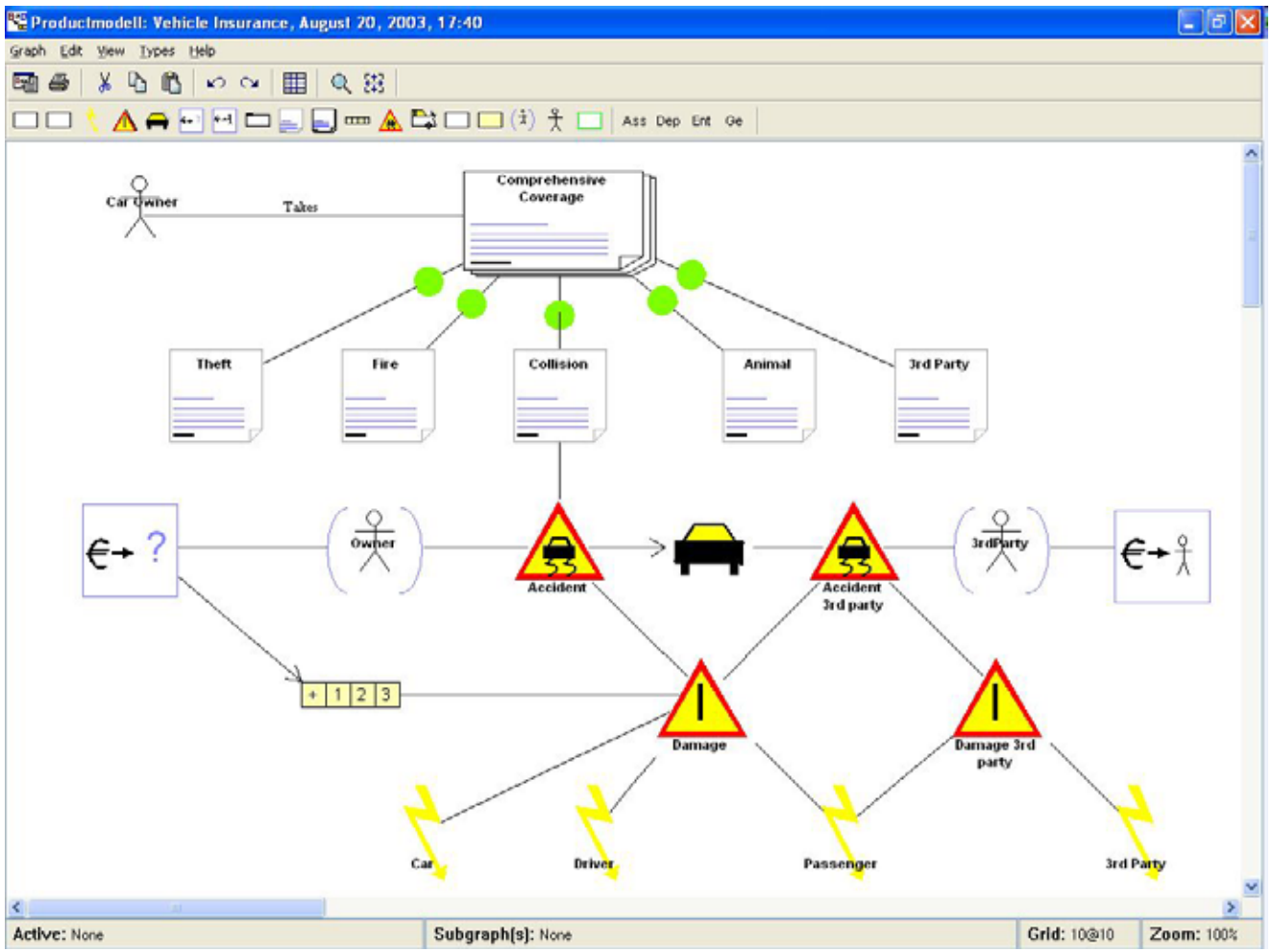
Generator definition



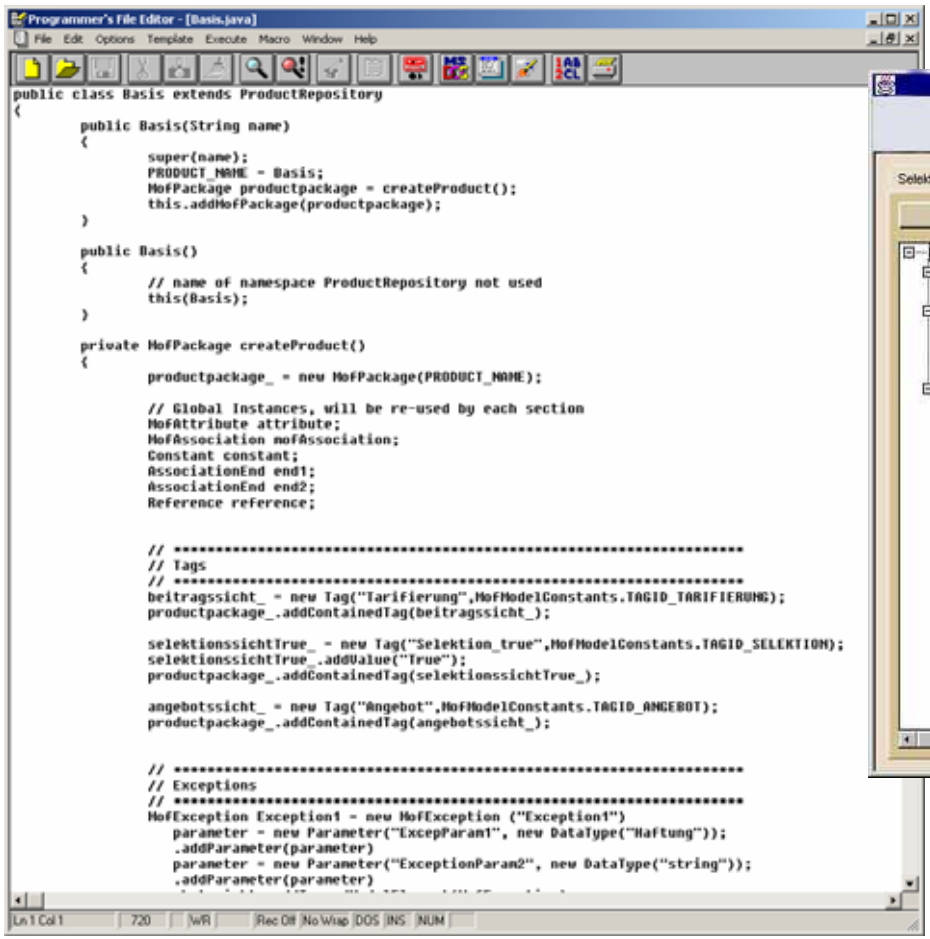
Generator output

```
def Note3_2227():
     appuifw.note(u"Registration made", 'conf')
     return Stop3_983

def Note3_6109():
     appuifw.note(u“SMS Cancellation sent", 'conf')
     return Stop3_983

def Note3_2543():
     appuifw.note(u“Conference registration: Welcome", ‘info')
     return Popup_menu3_2520

def Stop3_983():
# This applications stops here
     return appuifw.app.set_exit


...
```

Case2:保険のeコマースは、



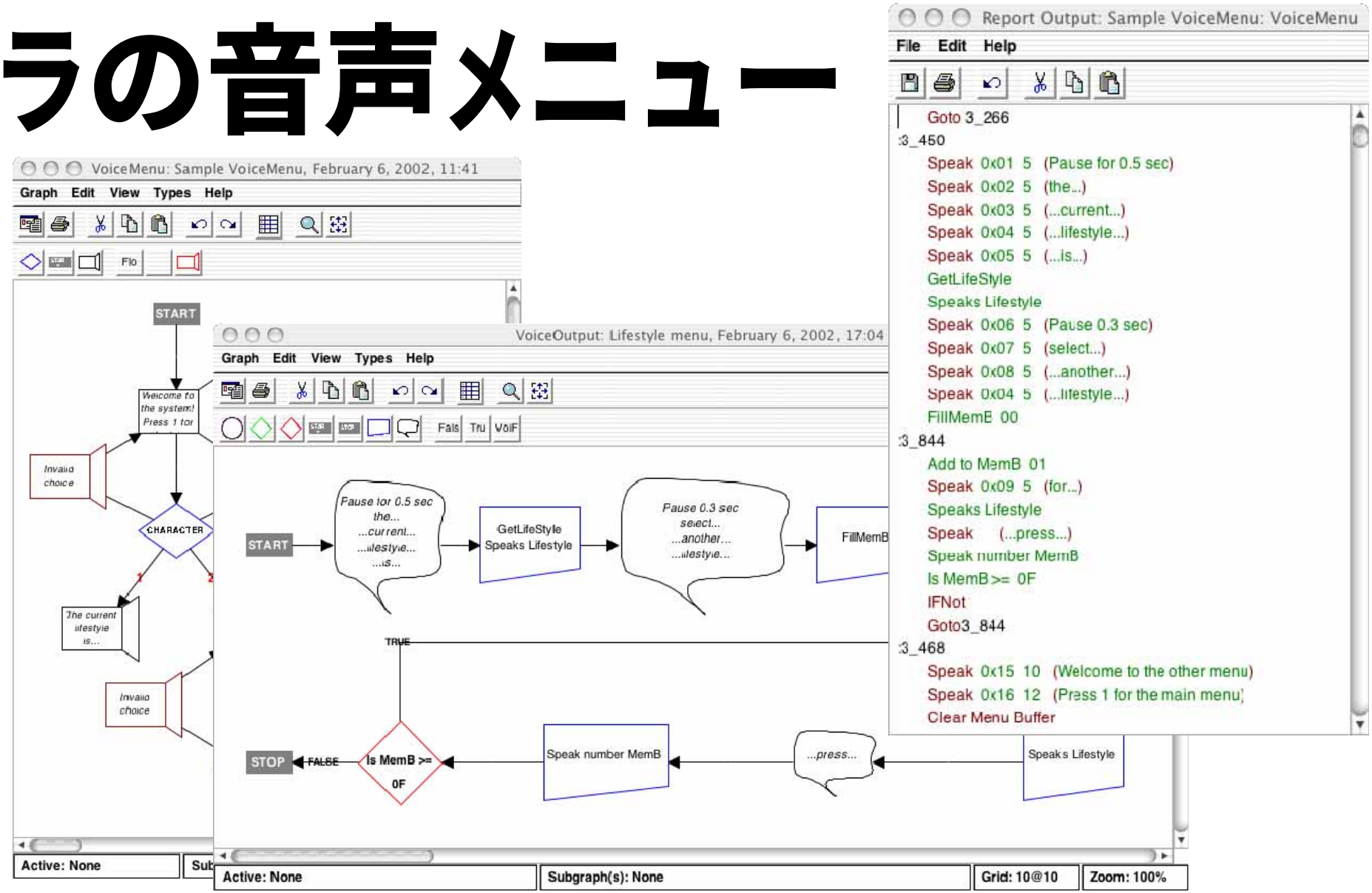
Java で書くより 3倍速くなった



Case3:ホームオートメーションの音声リモコン

- 8ビットマイコンへのアセンブラ
- 開発期間が1週間から1日

Case3: マイクロコントローラの音声メニュー



Matsushita Electric Works, Ltd.

“設計、テスト、修正を含めて6時間でドメイン固有の設定を行うことができました” Laurent Safa 氏

SIEMENS

“MetaEdit+ によるモデル化は、Eclipse に比べて非常に簡単で効果的であった“ Ulf Hesselbarth氏

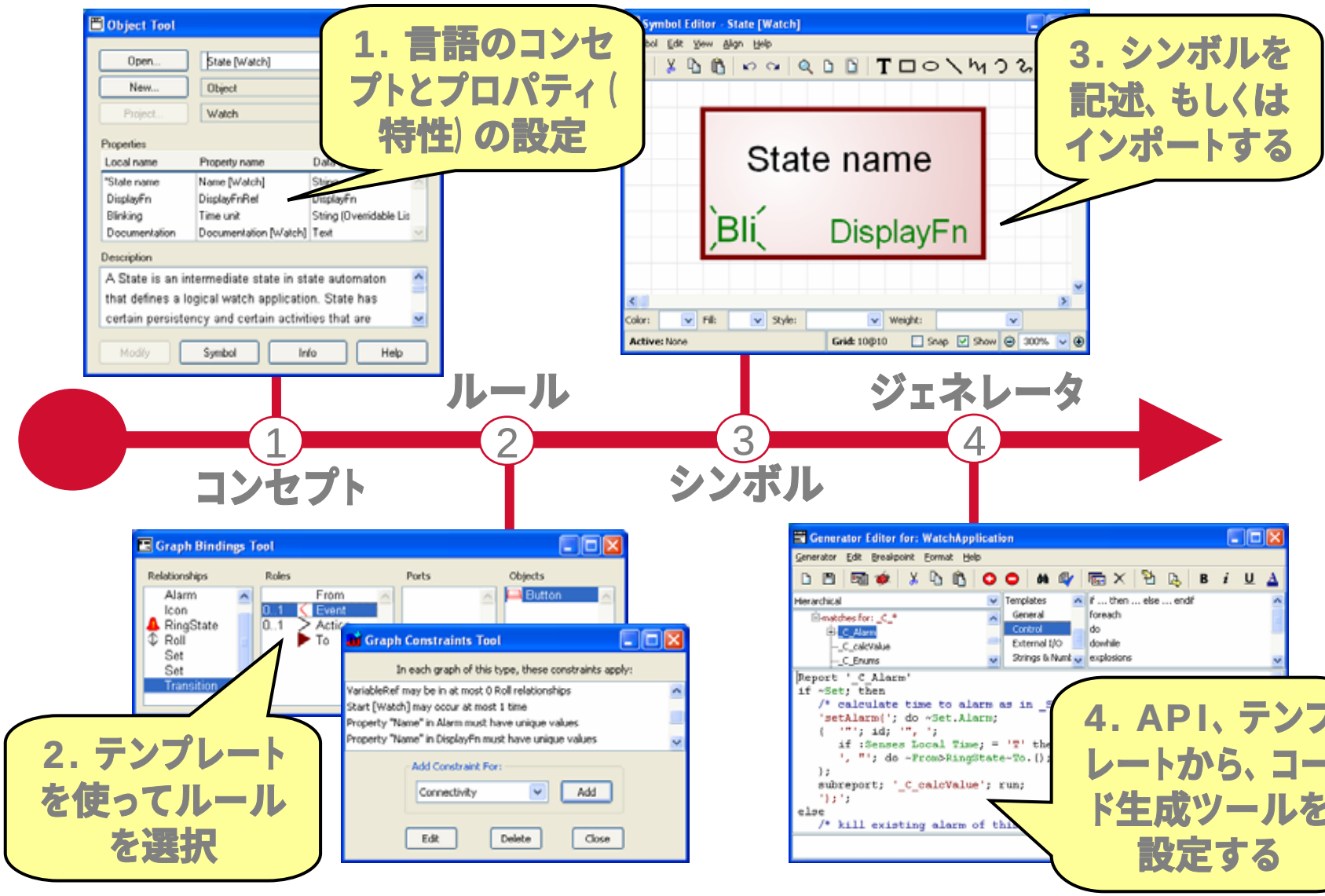
“群を抜いて優秀なDSLツールである” M.Voelter氏

NOKIA
CONNECTING PEOPLE

“非常に柔軟性があり、独自のコード生成ルールを簡単に定義することができた” D.Narraway氏

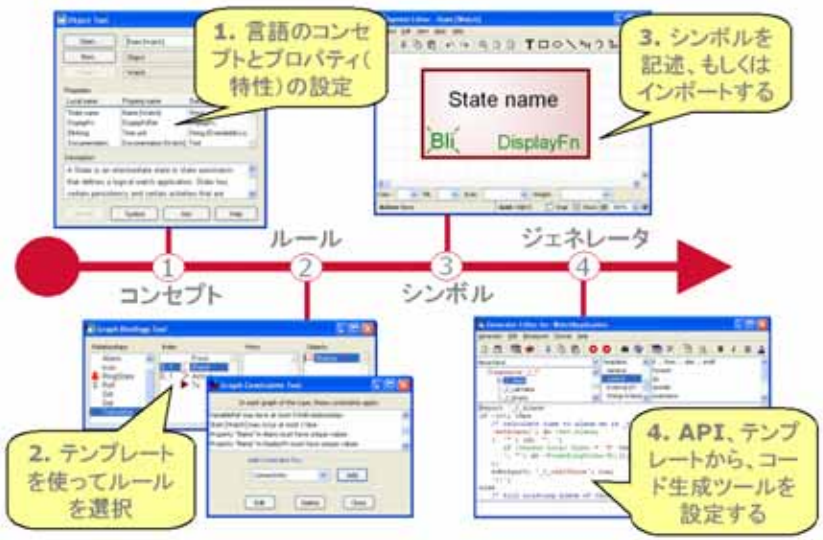
“最も洗練されたDSMツールです” Scott Ambler氏

DSMを作成するプロセス

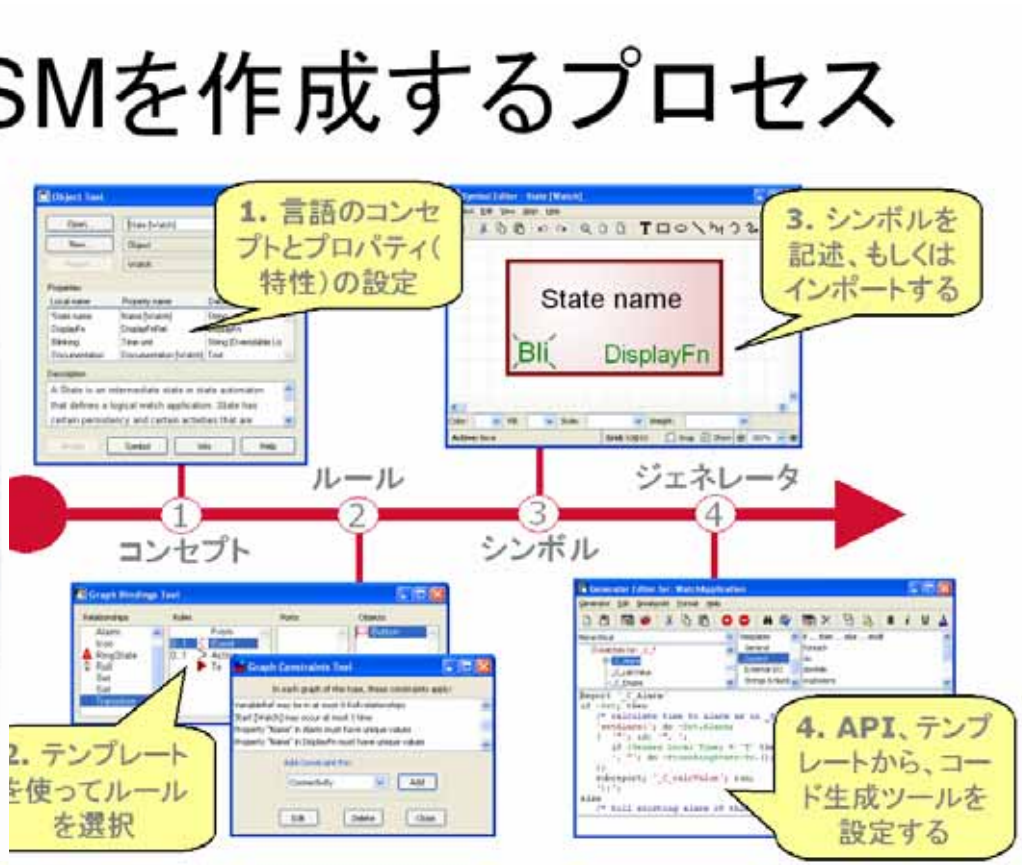


作成されるべきコードを
あらかじめ十二分に静的解析、単体テストすることで、

DSMを作成するプロセス



完成度の高い部品の組合せとしてシステムを構築できる



ありがとうございました。

FUJI SETSUBI