TOWARDS
Cluster in the Loop Simulation Framework based on Formal Model-based Testing for Embedded Network Systems

Eun-Hye Choi, Yoshiki Kinoshita, Hiroyuki Ozaki, Hayao Nakahara, AIST/CVS
Masahiro Aoki, Keiichi Yoshisaka, Hiroshi Mine, Daikin Industries, Ltd.
Toru Shimizu, Renesas Technology Corp.
Outline

• Backgrounds
  – Embedded network systems
  – Difficulties in V&V of embedded network systems
• Our testing framework
• Design of the proposed testing system
Embedded Network Systems

• Embedded systems all around us:
  – In-vehicle systems
  – Consumer electronics
  – Building and energy management systems.

Nowaday, they even form *networks*, which of course increase the scale and complexity a lot.

Verification and validation of embedded network systems wanted!
Outline

• Backgrounds

• Our testing framework
  – Our project
  – Target system
  – Current status of testing the target system
  – Key concept of our testing framework

• Design of the proposed testing system
Our Project

To develop a system to validate network system constituting air conditioners, especially for a large building.

Currently: feasibility study being done in collaboration with

- **Daikin** industries, ltd. (D):
  - developing air-conditioning systems and building energy management systems.
- **Renesas Technology** Corp. (R):
  - developing embedded micro processing units and related development tools.
Target System

Air-conditioner network for buildings developed by D.

- The target system consists of
  - controllers,
  - indoor/outdoor equipments including actuators
  - sensors, etc.
- Take the system construction where all MPUs are of R Ltd.
- Hundreds of nodes may be connected in the target system.
Current status of testing

• Specifications written in natural language, tables, and state transition diagrams.
• Manual generation of test cases.
• Manual testing using real equipments.
  → More testing is seeked for in validation of huge complex systems.
  → New testing method wanted.
New method for more test cases!

Current status of testing:
- Specifications written in natural language, tables, and state transition diagrams.
- Manual generation of test cases.
- Testing for many more test cases preferable
- New testing method wanted.
Two keys in our proposal

- Two Keys
  - Cluster in the Loop Simulation (CILS)
  - Formal Model-based Testing
Hardware in the Loop Simulation (HILS)

• HILS simulates the behavior of
  – a physical system and
  – an environment
  with which electronic control
  units of the embedded system
  interacts

• In HILS, both simulators and
  real equipments are used for
  testing.
  – helpful for testing complex
  real-time embedded sys.
  – Not suitable for testing large
  scale embedded network
  systems.
Cluster in the Loop Simulation (CILS)

- HILS uses hardware board; CILS uses a cluster instead.
- that simulates not only the physical system and the environment but also electronic control units to be connected them in a cluster computing system.

→ The whole embedded network system is simulated as software in a cluster system.
→ Real equipments are not necessary for testing.
## CILS vs. HILS

<table>
<thead>
<tr>
<th></th>
<th>CILS</th>
<th>HILS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application in design phase</td>
<td>😊</td>
<td>😞</td>
</tr>
<tr>
<td>Data processing</td>
<td>😊</td>
<td>😞</td>
</tr>
<tr>
<td>Real-time testing</td>
<td>😞</td>
<td>😊</td>
</tr>
<tr>
<td>Cost for testing</td>
<td>😊</td>
<td>😞</td>
</tr>
<tr>
<td>Time for testing</td>
<td>😊</td>
<td>😞</td>
</tr>
<tr>
<td>Scalability</td>
<td>😊</td>
<td>😞</td>
</tr>
<tr>
<td>Modification of network topology</td>
<td>😊</td>
<td>😞</td>
</tr>
<tr>
<td>Automatic confirmation of network configuration</td>
<td>😊</td>
<td>😞</td>
</tr>
</tbody>
</table>
Outline

• Backgrounds
• Our testing framework
  – Target system
  – Previous testing of the target system
  – Key concept of our testing framework
    • CILS
    • Formal Model-based Testing
• Design of the proposed testing system
Model-based Testing

Generating a test suite from a formal specification (model).

- Requirements
- Modeling
- Model (formal spec.)
- Test case generation
- Test suite (Inputs)
- Testing
- Test Output
- Test result assessment
- Expected Input / Output relation
- Testing Result
Model-based Testing

• There are many different ways
  – What kind of model description is supposed?
    • Finite State Machines: Decision tables, State charts..
  – How a test suite is derived?
    • Searching for execution traces in an abstract model,
    • Finding test cases by constraint programming,
    • Using counterexample paths by model checking,
    • Selecting test cases by partitioning a model to classes using theorem proving…
Our Approach

• Given informal specification written in Japanese etc.,
• Derive a formal specification (Model) written in Agda.
• Build a tool which automatically generates test suites for the specification.
  – The specification must be in a form such that such automatic generation is possible.
A case study
Informal specification I

1. 目的
冷房モード又は除湿モード時の、室内換気系統を遮断するために、熱交換が断定温度以下に下がりすぎないように、回転扇風機を遮断させる。

2. 入出力情報

<table>
<thead>
<tr>
<th>入力ビーム</th>
<th>出力ビーム</th>
</tr>
</thead>
<tbody>
<tr>
<td>①冷房扇風状態</td>
<td>①冷房防止ステータス</td>
</tr>
<tr>
<td>②室内熱交換温度</td>
<td>②室内ファン負荷回転数</td>
</tr>
<tr>
<td>③冷房モード</td>
<td></td>
</tr>
<tr>
<td>④除湿</td>
<td></td>
</tr>
<tr>
<td>⑤除湿制御指令</td>
<td></td>
</tr>
<tr>
<td>⑥屋外換気状態</td>
<td></td>
</tr>
</tbody>
</table>

3. 現象説明

(1) 強制換気以外の制御内容
- 冷房扇風モードが熱交-ドライモード[b]時に冷房防止ステータスを決定する。
- 室内換気状態に換気タイマーTTOが設定されている場合、熱交換が断定温度以下に下がりすぎると、冷房防止ステータスが複数ゾーンとする。[TTO:換気制御タイマー]
- TTOは解除換気停止でタイマークリアする。
- TTOオーバー後は、以下の熱交換温度ゾーンによって熱交換防止ステータスを決定する。

<table>
<thead>
<tr>
<th>熱交換温度ゾーン</th>
<th>凍結防止ステータス</th>
</tr>
</thead>
<tbody>
<tr>
<td>冷房ゾーン</td>
<td>冷房防止ステータス</td>
</tr>
<tr>
<td>ドライゾーン</td>
<td></td>
</tr>
<tr>
<td>セーラドライモード</td>
<td></td>
</tr>
</tbody>
</table>

※13 熱交換温度ゾーンがZゾーン以外からZゾーンに変化した場合、室温調整は至らないため、

※14 熱交換温度ゾーンがZゾーンからZゾーンに変化した場合、室温調整は至らない。熱交換温度ゾーンがZゾーンからZゾーンに変化した場合、室温調整は至らない。

※23 一旦、冷房防止ステータスが停止になったら、熱交換温度ゾーンがZゾーンになるまで停止とする。

2006年3月6日
Informal specification II
Formalisation

Following tasks, for instance, are needed during the process of formalisation of the specification.

• Clarification of variables and range of values.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Range of Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1 (state_compressor)</td>
<td>ON, OFF</td>
</tr>
<tr>
<td>v6 (state_antifreeze)</td>
<td>comeback, up, same, pendency, abort</td>
</tr>
<tr>
<td>v10 (state_heat-exchange-temp)</td>
<td>A,B,C,D,E</td>
</tr>
<tr>
<td>tto</td>
<td>clearstart, over</td>
</tr>
</tbody>
</table>

• Clarification of the relations of variables.
  – Previous v1=OFF and v1=ON then v6=comeback and tto=clearstart.
  – tto=over and v10=A then v6=comeback.
  – …
Formal model in Agda I

module 20091029aircond-e where
open import Logic
open import Data.Nat hiding (_<_ ; _>_ ; _+_) open import Data.Fin hiding (_<_ ; _+_) 
data ComperssorOpStateType : Set where
  ON : ComperssorOpStateType
  OFF : ComperssorOpStateType

data TemperatureType : Set where
  -32'0℃ : TemperatureType
  -32'5℃ : TemperatureType
  -- 0.5 inteval
  +95'5℃ : TemperatureType

data OpModeType : Set where
  StopMode : OpModeType
  CoolMode : OpModeType
  WarmMode : OpModeType
  DryMode : OpModeType
  ReheatDryMode : OpModeType

data ExteriorFreePreventionType : Set where
  Normal : ExteriorFreePreventionType
  FreezePrevention : ExteriorFreePreventionType

data FreezePreventionStateType : Set where
  Return : FreezePreventionStateType
  Up : FreezePreventionStateType
  NoChange : FreezePreventionStateType
  Droop : FreezePreventionStateType
  Stopped : FreezePreventionStateType

data RevolutionType : Set
  RevolutionType = Fin 150 -- ×10 rpm

data HEXTemperatureZoneType : Set where
  A : HEXTemperatureZoneType
  B : HEXTemperatureZoneType
  C : HEXTemperatureZoneType
  D : HEXTemperatureZoneType
  E : HEXTemperatureZoneType
data InHEXTemperatureStateType : Set where
  GoingDown : InHEXTemperatureStateType
  GoingUp : InHEXTemperatureStateType
record State : Set where
  field
    CompressorOpState : CompressorOpStateType -- input
    InteriorHEXTemperature : TemperatureType -- input
    OpMode : OpModeType -- input
    ExteriorFreezePrevention :
      ExteriorFreezePreventionType -- input
    FreezePreventionState :
      FreezePreventionStateType -- output
    InteriorFanRevolutionGoalGLB : RevolutionType -- output
  InteriorHEXTemperatureState :
    InHEXTemperatureStateType
  HEXTemperatureZone : HEXTemperatureZoneType
  tx : ℕ -- internal variable
  tto : ℕ -- internal variable

postulate
  ttofg : ℕ
  ttofgktr : ℕ
  ttofgdry : ℕ
  dto0 : TemperatureType
  dto1 : TemperatureType
  dto2 : TemperatureType
  dto3 : TemperatureType
  dto4 : TemperatureType
  dtktr0 : TemperatureType
  dtktr1 : TemperatureType
  dtktr2 : TemperatureType
  dtktr3 : TemperatureType
  dtktr4 : TemperatureType
Formal model in Agda III

\begin{align*}
v1 &= \text{State.CompressorOpState} \\
v2 &= \text{State.InteriorHEXTemperature} \\
v3 &= \text{State.OpMode} \\
\quad \text{-- } v4 &= \text{State.室温} \\
v5 &= \text{State.ExteriorFreezePrevention} \\
v6 &= \text{State.FreezePreventionState} \\
\quad \text{-- } v7 &= \text{State.InteriorFanApproxRevolution} \\
v8 &= \text{State.InteriorFanRevolutionGoalGLB} \\
v9 &= \text{State.InteriorHEXTemperatureState} \\
v10 &= \text{State.HEXTemperatureZone} \\
\end{align*}

\begin{align*}
\text{postulate} \\
W2 : \text{RevolutionType} & \text{ -- constant!} \\
\text{TTO} : \mathbb{N} \\
\text{DC1} : \text{TemperatureType} \\
\text{DC2} : \text{TemperatureType} \\
\text{DC3} : \text{TemperatureType} \\
\text{DC4} : \text{TemperatureType} \\
\text{DC5} : \text{TemperatureType} \\
\end{align*}

\begin{align*}
\text{postulate} \\
\text{pre} : \text{State} & \text{ -- previous state} \\
\text{current} : \text{State} & \text{ -- current state} \\
\text{inAzone} : \text{HEXTempreatureZoneType} & \to \text{Pr} \\
\text{inBzone} : \text{HEXTempreatureZoneType} & \to \text{Pr} \\
\text{inCzone} : \text{HEXTempreatureZoneType} & \to \text{Pr} \\
\text{inDzone} : \text{HEXTempreatureZoneType} & \to \text{Pr} \\
\text{inEzone} : \text{HEXTempreatureZoneType} & \to \text{Pr} \\
\text{Counting} : \mathbb{N} & \to \text{Pr} \\
\text{CountOver} : \mathbb{N} & \to \text{Pr} \\
\text{Start} : \text{Pr} & \to \text{Pr} \\
\text{ClearStart} : \mathbb{N} & \to \text{Pr} \\
\text{Stop} : \mathbb{N} & \to \text{Pr} \\
\_\_ \geq \_ : \text{TemperatureType} & \to \text{TemperatureType} & \to \text{Pr} \\
\_\_ \leq \_ : \text{TemperatureType} & \to \text{TemperatureType} & \to \text{Pr} \\
\_\_ > \_ : \text{TemperatureType} & \to \text{TemperatureType} & \to \text{Pr} \\
\_\_ + \_ : \text{TemperatureType} & \to \text{TemperatureType} & \to \text{TemperatureType} \\
\end{align*}
Formal model in Agda IV

prop1 : Pr
prop1 = v1 pre ≡ OFF ∧ v1 current ≡ ON
    → v6 current ≡ Return ∧ ClearStart (State.tto current)

prop2 : Pr
prop2 =
    CountOver (State.tto current) ∧ inAzone (v10 current)
    → v6 current ≡ Return

prop3 : Pr
prop3 =
    CountOver (State.tto current) ∧ inBzone (v10 current)
    → v6 current ≡ Up

prop4 : Pr
prop4 =
    CountOver (State.tto current) ∧ inCzone (v10 current)
    → v6 current ≡ NoChange

prop5 : Pr
prop5 =
    CountOver (State.tto current) ∧ inDzone (v10 current)
    → v6 current ≡ Droop

prop6 : Pr
prop6 = CountOver (State.tto current) ∧ inEzone (v10 current)
    ∧ Counting (State.tx current)
    → v6 current ≡ Droop

prop7 : Pr
prop7 = CountOver (State.tto current) ∧ inEzone (v10 current)
    ∧ CountOver (State.tx current)
    → v6 current ≡ Stopped

prop7-1 : Pr
prop7-1 = Counting (State.tto current)
    → v6 current ≡ Return

prop8 : Pr
prop8 = inDzone (v10 pre) ∧ inEzone (v10 current)
    → ClearStart (State.tx current)

prop9 : Pr
prop9 = inEzone (v10 pre) ∧ inDzone (v10 current)
    ∧ Counting (State.tx current)
    → Stop (State.tx current)
Formal model in Agda V

\[ \text{prop10} \equiv \text{Pr} \]
\[ \text{prop10} = \text{v1 pre} \equiv \text{ON} \land \text{v1 current} \equiv \text{OFF} \]
\[ \rightarrow \text{Stop (State.tto current)} \]

\[ \text{prop11} \equiv \text{Pr} \]
\[ \text{prop11} = \]
\[ \text{v6 pre} \equiv \text{Stopped} \land \lnot \text{inAzone (v10 pre)} \]
\[ \rightarrow \text{v6 current} \equiv \text{Stopped} \]

\[ \text{prop12} \equiv \text{Pr} \]
\[ \text{prop12} = \text{v6 current} \equiv \text{Stopped} \]
\[ \rightarrow \text{v8 current} \equiv \text{W2} \]

\[ \text{prop13} \equiv \text{Pr} \]
\[ \text{prop13} = \text{v1 pre} \equiv \text{OFF} \land \text{v1 current} \equiv \text{ON} \]
\[ \rightarrow \text{v8 current} \equiv \text{zero} \]

\[ \text{prop14} \equiv \text{Pr} \]
\[ \text{prop14} = \text{v9 current} \equiv \text{GoingDown} \land \text{v2 current} \geq \text{DC2} \]
\[ \rightarrow \text{inAzone (v10 current)} \]

\[ \text{prop15} \equiv \text{Pr} \]
\[ \text{prop15} = \]
\[ \text{v9 current} \equiv \text{GoingDown} \land \text{v2 current} \geq \text{DC3} \]
\[ \land \text{v2 current} < \text{DC2} \]
\[ \rightarrow \text{inBzone (v10 current)} \]

\[ \text{prop16} \equiv \text{Pr} \]
\[ \text{prop16} = \text{v9 current} \equiv \text{GoingDown} \land \text{v2 current} \geq \text{DC4} \land \text{v2 current} < \text{DC3} \]
\[ \rightarrow \text{inCzone (v10 current)} \]

\[ \text{prop17} \equiv \text{Pr} \]
\[ \text{prop17} = \text{v9 current} \equiv \text{GoingDown} \land \text{v2 current} \geq \text{DC5} \land \text{v2 current} < \text{DC4} \]
\[ \rightarrow \text{inDzone (v10 current)} \]

\[ \text{prop18} \equiv \text{Pr} \]
\[ \text{prop18} = \text{v9 current} \equiv \text{GoingDown} \land \text{v2 current} < \text{DC5} \]
\[ \rightarrow \text{inEzone (v10 current)} \]

\[ \text{prop19} \equiv \text{Pr} \]
\[ \text{prop19} = \text{v9 current} \equiv \text{GoingUp} \land \text{v2 current} \geq \text{DC1} \]
\[ \rightarrow \text{inAzone (v10 current)} \]

\[ \text{prop20} \equiv \text{Pr} \]
\[ \text{prop20} = \text{v9 current} \equiv \text{GoingUp} \land \text{v2 current} \geq \text{DC2} \land \text{v2 current} < \text{DC1} \]
\[ \rightarrow \text{inBzone (v10 current)} \]
Formal model in Agda VI

prop21 : Pr
prop21 = v9 current ≡ GoingUp
   ∧ v2 current >= DC3 ∧ v2 current < DC2
   → inCzone (v10 current)

prop22 : Pr
prop22 = v9 current ≡ GoingUp
   ∧ v2 current >= DC5 ∧ v2 current < DC3
   → inDzone (v10 current)

prop23 : Pr
prop23 = v9 current ≡ GoingUp ∧ v2 current < DC5
   → inEzone (v10 current)

prop24 : Pr
prop24 = v3 current ≡ ReheatDryMode
   → DC1 ≡ dtktr4 ∧ DC2 ≡ dtktr1 + dtktr3
   ∧ DC3 ≡ dtktr1 + dtktr2
   ∧ DC4 ≡ dtktr1 ∧ DC5 ≡ dtktr0

prop25 : Pr
prop25 = v3 current ≡ CoolMode ∨ v3 current ≡ DryMode
   → DC1 ≡ dto4 ∧ DC2 ≡ dto1 + dto3
   ∧ DC3 ≡ dto1 + dto2
   ∧ DC4 ≡ dto1 ∧ DC5 ≡ dtktr0

prop26 : Pr
prop26 = v2 pre ≡ v2 current → v9 current ≡ v9 pre

prop27 : Pr
prop27 = v2 pre < v2 current → v9 current ≡ GoingUp

prop28 : Pr
prop28 = v2 pre > v2 current → v9 current ≡ GoingDown
Test case

- Test case: A state $s$ (or a sequence of states) in the state transition system $S$ representing the Agda model.
- Expected output: $s'$ such that transition $(s,s') \in S$.
- Ex 3.

Test suite (Inputs)

<table>
<thead>
<tr>
<th>v1</th>
<th>v6</th>
<th>v10</th>
<th>tto</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON</td>
<td>*</td>
<td>*</td>
<td>clearstart</td>
</tr>
<tr>
<td>ON</td>
<td>comeback</td>
<td>A</td>
<td>over</td>
</tr>
<tr>
<td>ON</td>
<td>up</td>
<td>B</td>
<td>over</td>
</tr>
<tr>
<td>ON</td>
<td>same</td>
<td>C</td>
<td>over</td>
</tr>
<tr>
<td>ON</td>
<td>pendency</td>
<td>D</td>
<td>over</td>
</tr>
<tr>
<td>OFF</td>
<td>*</td>
<td>*</td>
<td>clearstart</td>
</tr>
<tr>
<td>OFF</td>
<td>comeback</td>
<td>A</td>
<td>over</td>
</tr>
<tr>
<td>OFF</td>
<td>up</td>
<td>B</td>
<td>over</td>
</tr>
<tr>
<td>OFF</td>
<td>same</td>
<td>C</td>
<td>over</td>
</tr>
<tr>
<td>OFF</td>
<td>pendency</td>
<td>D</td>
<td>over</td>
</tr>
</tbody>
</table>

Expected Outputs

<table>
<thead>
<tr>
<th>v1</th>
<th>v6</th>
<th>v10</th>
<th>tto</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>*</td>
<td>*</td>
<td>clearstart</td>
</tr>
<tr>
<td>*</td>
<td>comeback</td>
<td>A</td>
<td>over</td>
</tr>
<tr>
<td>*</td>
<td>up</td>
<td>B</td>
<td>over</td>
</tr>
<tr>
<td>*</td>
<td>same</td>
<td>C</td>
<td>over</td>
</tr>
<tr>
<td>*</td>
<td>pendency</td>
<td>D</td>
<td>over</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>v1</th>
<th>v6</th>
<th>v10</th>
<th>tto</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON</td>
<td>comeback</td>
<td>*</td>
<td>clearstart</td>
</tr>
<tr>
<td>OFF</td>
<td>*</td>
<td>*</td>
<td>clearstart</td>
</tr>
<tr>
<td>OFF</td>
<td>comeback</td>
<td>A</td>
<td>over</td>
</tr>
<tr>
<td>OFF</td>
<td>up</td>
<td>B</td>
<td>over</td>
</tr>
<tr>
<td>OFF</td>
<td>same</td>
<td>C</td>
<td>over</td>
</tr>
<tr>
<td>OFF</td>
<td>pendency</td>
<td>D</td>
<td>over</td>
</tr>
</tbody>
</table>
Test case generation

• Test case generation from a formal specification by hand (for this case study.)
  Automatic generation is aimed at.

• Test case for a spec is a valuation of variables in the spec., which makes the required properties true.
  Some variables are input variables, others are output variables.

• # of valid test cases will be enormous
  ➔ boundary conditions to the required properties.
Benefits of our Framework

• By Formal Model-based Testing
  → Automation of test case generation,
    Improvement of quality of test cases.
  – In the previous work with Company R, we confirmed
    the improvement of test case quality by the model-
    based test generation using Agda.

• By Cluster in the Loop Simulation (CILS)
  → Automation of testing,
    Expansion of scalability,
    Reducing time for testing.
Outline

• Backgrounds
• Our testing framework
• Design of the proposed testing system
Outline Design of the proposed CILS System

Specifications of the target system

Formal Model-based Testing

Cluster Simulator

- HW simulator
- Software
- Environment Simulator
- Network Simulator

Testing Tool

- I/O interface for test cases
- Testing result judging program

Test case generation tool

- Testing strategy
- Model-based test case generation program

Formal Modeling

- Agda model of system spec.
- Agda model of environment spec.
Outline Design
of the proposed CILS System

Specifications of the target system

Formal Model-based Testing

Test case generation tool
- Testing strategy
- Model-based test case generation program

Testing Tool
- I/O interface for test cases
- Testing result judging program

Cluster Simulator
- HW simulator
- Software
- Environment Simulator
- Network Simulator

Formal Modeling
- Agda model of system spec.
- Agda model of environment spec.
Formal Modeling

Given informal specifications

- **System specification:**
  - natural language text or state transition diagrams.
- **Environment specification:**
  - e.g., relations between temp. and airflow
  - domain specific knowledge.

Formal Description in Agda

- system spec.
- environment spec.
Outline Design of the proposed CILS System

Specifications of the target system

Formal Model-based Testing

- Formal Modeling
  - Agda model of system spec.
  - Agda model of environment spec.

- Test case generation tool
  - Testing strategy
  - Model-based test case generation program

- Testing Tool
  - Testing Monitor
  - I/O interface for test cases
  - Testing result judging program

Cluster simulation

- Cluster Simulator
  - HW simulator
  - HW simulator
  - Software
  - Software
  - Environment Simulator
  - Network Simulator
Formal Model-based Testing

Test case generation tool

Testing strategy

Model-based test case generation program

Test Suite

Testing Tool

Testing Monitor

I/O interface for test cases

Testing result judging program

Agda Model

Test case

Testing result
Outline Design of the proposed CILS System

Specifications of the target system

Formal Model-based Testing

Cluster simulation

Formal Modeling
- Agda model of system spec.
- Agda model of environment spec.

Test case generation tool
- Testing strategy
- Model-based test case generation program

Testing Tool
- Testing Monitor
- I/O interface for test cases
- Testing result judging program

Cluster Simulator
- HW simulator
- HW simulator
- Software
- Software
- Environment Simulator
- Network Simulator
• The simulators will be developed in the cluster system called SATSUKI in Collaborative Facilities for Verification in AIST.
Our Position

• Three parties got together in June 2009.
  – Feasibility study till March 2010, seeking for support.
  – Start the real project from April 2010, hopefully.

• Planned:
  – CILS based on formal-model based testing
  – Testing system for air-conditioner with automatic test case generator
  – 3 year project proposal.

• Issues to be detailed
  – Method for test case generation
    what should be boundary conditions?
  – Framework for verification (esp. need for temporal properties?)
  – Fast simulators