Application of a Formal Specification Language in the Development of the “Mobile FeliCa” IC Chip Firmware for Embedding in Mobile Phone

Formal Methods 2008
15th International Symposium on Formal Methods
May 26-30, 2008
Turku, Finland

FeliCa Networks, Inc. and Sony Corporation
Agenda

• What is “Mobile FeliCa”

• Highlight Results

• Summary and Future Issues
What is “Mobile FeliCa”

• “FeliCa” is a **contactless IC card technology** widely used in Japan.

• FeliCa is developed and promoted by Sony Corporation.

• FeliCa is used for electric money, train tickets, identifications, door keys and so on.

• Today, “Mobile FeliCa” IC chips are **embedded in over 50 million mobile phones.**
Project Duration and Members

- The project duration was **three years and three months**. It finished **on schedule**.

- There were **50-60 members**. The average age was about **30 years old**.

- We had **no knowledge and no experience with formal methods**.
We organized three teams:

- **Specification** Team (5-20 members)
- **Firmware Implementation** Team (15-20 members)
- **Testing** Team (25-35 members)
Results

• 383 pages of a protocol manual written in the natural language

• 677 pages of an external specification document written in the formal specification language

• We developed executable formal specifications.

• Our formal specifications are about 100,000 LOC including test cases (about 60,000 LOC) and comments.

• Using this specifications, we implemented the C++ code of about 110,000 LOC, inclusive of comments.
The average productivity of VDM++ code for the formal specifications was about 1,900 LOC per engineer per month.
Formal Specification Errors in Specification Phase

<table>
<thead>
<tr>
<th>Phase of Development Process</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Describing Specifications</td>
<td>162</td>
</tr>
<tr>
<td>Executing and Unit Testing Specifications</td>
<td>116</td>
</tr>
<tr>
<td>Reviewing Specifications</td>
<td>93</td>
</tr>
<tr>
<td>Communicating with Firmware Engineers</td>
<td>69</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>440</strong></td>
</tr>
</tbody>
</table>

Debug Density = 440/40,000 = about 11 errors/kLOC

The formal method contributes to enhancing the quality of deliverables at the early stage of development process.
## Errors in Firmware Implementation Phase

<table>
<thead>
<tr>
<th>Reason for Errors</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing description</td>
<td>0.2%</td>
</tr>
<tr>
<td>Erroneous description</td>
<td>0%</td>
</tr>
<tr>
<td>Unclear description</td>
<td>1.8%</td>
</tr>
<tr>
<td>Oversight</td>
<td>5.6%</td>
</tr>
<tr>
<td>Insufficient understanding</td>
<td>10.7%</td>
</tr>
<tr>
<td>Insufficient confirmation</td>
<td>0%</td>
</tr>
<tr>
<td>Failure of change propagation</td>
<td>0.2%</td>
</tr>
<tr>
<td>Others (reasons unrelated to specifications)</td>
<td>81.5%</td>
</tr>
</tbody>
</table>
It can be said that we have successfully described the specifications in a precise way.

The formal methods are useful for finding errors in the early stages of development.
Errors in Firmware Implementation Phase

<table>
<thead>
<tr>
<th>Reason for Errors</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unclear description</td>
<td>1.8%</td>
</tr>
<tr>
<td>Oversight</td>
<td>5.6%</td>
</tr>
<tr>
<td>Insufficient understanding</td>
<td>10.7%</td>
</tr>
<tr>
<td>Insufficient confirmation</td>
<td>0%</td>
</tr>
<tr>
<td>Failure of change propagation</td>
<td>0.2%</td>
</tr>
<tr>
<td>Others (reasons unrelated to specifications)</td>
<td>81.5%</td>
</tr>
</tbody>
</table>

• On the other hand, the total percentage of “oversight” errors and “insufficient understanding” errors was 16.3%.

• This was due to the fact that the separations between the actual specifications and the code required to execute the specifications was unclear.

Future Issue: Easy-to-read
Test Environment
Executable formal specifications, firmware on development board and IC chips were automatically tested using the test environment and test scripts.
Test Results

- The line coverage rate of the formal specifications by black-box testing and visual inspection was 100%.

- “Random Test” is a continuous test.
- The test tool sends random commands continuously to the test target and checks whether the test target sends back correct responses.
- Random test tools is generated from VDM++ model.

- By carrying out about 7,000 black-box tests and 100 million random tests, the high quality of IC chips was achieved.
Summary

- We have developed the “Mobile FeliCa” IC Chip firmware specifications with a formal method using VDM++ and VDMtools.

- We use Lightweight Formal Methods.

- We described and tested formal specifications without proof.

- Thanks to the formal methods, there are no problems related to the software specifications since the first release.
We are now developing the next generation of specifications using VDM++ and VDMTools at Sony.

- Validating whether specifications fulfill requirements.
- Managing variations for the product line.
- Validation and testing of the formal specifications; for example validation of whether a security specification is logically consistent.
- Framework for describing specifications that are easy-to-read and executable.
Any Questions?

Thank you!