

Transacted Memory for Smart Cards

Pieter Hartel, Univ. Twente, NL

Michael Butler, Univ. Southampton, UK

Eduard de Jong, Sun Micro systems, USA

Mark Longley, Univ. Southampton, UK

Overview

- What is transacted memory?
- Combining Z, Promela (SPIN) and C
- Conclusions

Java Card transactions

- atomic updates
- audit trail
- limited resources
(64KB ROM, 64KB EEPROM, 2KB RAM)
- Java Card logs previous value

Transacted memory – abstract

Before	After
tag 1, gen 2	unused
10£, Feb 3, 2001	
tag 1, gen 3	tag 1, gen 3
80£, Feb 5, 2001	80£, Feb 5, 2001
unused	tag 1, gen 4
	70£, Feb 8, 2001
tag 5, gen 1	tag 5, gen 1
10 Downing street	10 Downing street

Write, commit and verify:

```
info = { 70£, Feb 8, 2001 };
```

```
tag = 1;
```

```
Write( tag, info );
```

```
Commit( tag );
```

```
assert( DRead( tag ) == info );
```

Transacted memory – concrete

Before	After
unused	tag 1, gen 4, page 0
	70£
tag 1, gen 4, page 2	tag 1, gen 4, page 2
2001	2001
tag 1, gen 4, page 1	tag 1, gen 4, page 1
Feb 8	Feb 8

Write:

```
info = { 70£, Feb 8, 2001 };
```

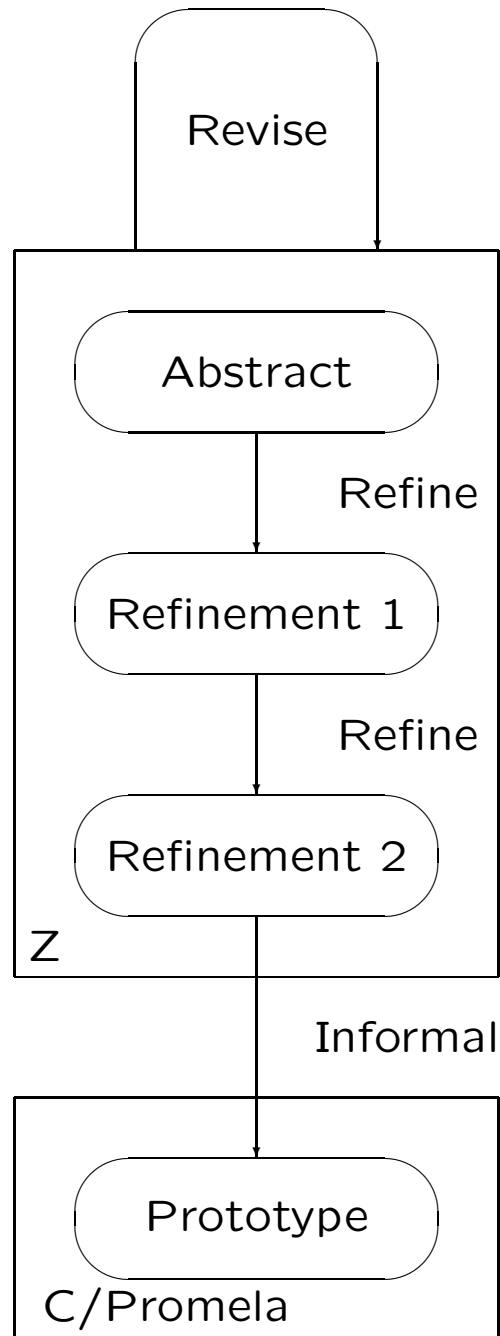
```
tag = 1;
```

```
Write_page( tag, 2, info[2]);
```

```
Write_page( tag, 1, info[1]);
```

```
Write_page( tag, 0, info[0]);
```

Methodology



Functions in C and Promela

Function

Call

```
byte NewTag(byte size) {  
    byte tag;  
    ...  
    return tag;  
}
```

```
byte tag = NewTag( 3 );
```

Promela: Non-deterministic choice

```
proctype NewTag() {  
    byte size, tag;  
    ....  
    go?MSize,size ->  
    .....  
    if  
    :: done!Mabort; ...  
    :: done!MTag,tag; ...  
    fi
```

```
    byte tag;  
    go!MSize,3;  
  
    if  
    :: done?Mabort -> ...  
    :: done?MTag,tag -> ...  
    fi
```

Test program

- 2000 page writes
- 65 aborted writes
- No failed assertions

Issues – Z

- Non-standard constructs

```
| PROCEDURE  
| release : Memory ×  $\mathbb{P}$  Loc → Memory  
| release(mem, Iset) =  
|           FOR I IN Iset DO write(mem, I, ...)
```

- Syntax and typing problems, e.g.
 $a \times b$ instead of $a * b$
- 10 issues in 2 pages

Issues – Promela/C

- Typing problems, missing definitions
- Committed write does not commit
- Uncommitted pages not released
- Data may be committed twice

Conclusions

- Using non-standard Z does not help
- Z good for abstract
- Promela good for concrete
- ad-hoc common notation
- Prototype is RAM space efficient
- Time efficient with HW support?

More work on Transacted Memory

Erik Poll, Univ. Nijmegen, NL

Using **JML** & **Java** instead of **Z** & **C**:

- translation of abstract Z spec to JML
- translation of C implementation to Java

Both relatively straightforward.

JML vs Z specification

– Z easier to read than JML

+ JML spec can easily be made executable

Java vs C implementation

- C closer to realistic machine-code implementation
- + Java implementation of enumerations as classes revealed bug
(but is clumsy; type-safe enums would be nicer!)
- + Java's exception mechanism enables realistic test scenarios, including card tears

```
public Tag NewTag(byte size) throws CardTearException {  
    ...  
}
```

Java & JML vs Z & C

+ abstract JML spec and concrete Java implementation
in the same language (namely Java)

Future work: relating abstract spec and implementation