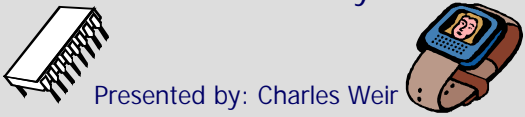


Patterns for Small Machines

NOOPS Presentation Using Objects in Systems with Limited Memory



Presented by: Charles Weir
Co-author: James Noble


Small Memory Software

- There will always be niches for small memory software!
 - Mobile phones, palmtops
 - Embedded or Intravenous systems
 - Backwards compatibility
 - Small machines are cheaper
 - less memory to pay for
 - Running 1000 sessions in a 1 GB server
 - 1M per program is a small machine

Overview

- **Fixed Data Structure**
- Partial Failure
- Multiple Representation
- Packed Data

The StrapItOn System



- A portable, *wrist-mounted PC*
- Cheap, low power,
- fully functional
- **But: Only 1M memory (+ 2M Rom)**
- O/S functions (persistent storage, communications, internet, I/O)
- Application programs?
- Will the software fit the space available?

StrapItOn Priority Messages

- StrapItOn can receive high priority messages via cellular network
- These messages are short (~ 100 characters)
- The messages do not arrive often
- Priority Messages must be processed quickly and reliably
- Failure to handle a message is unacceptable

Priority Messages Resolved

- After consultation with users and PMS protocol designers, the StrapItOn developers agreed:
- Messages would be strictly limited to 128 8-bit characters
 - Excess characters can be truncated
- The StrapItOn wristtop needs to store only five messages at any given time
 - When a new message arrives, the oldest message can be deleted.
- Messages stored in an 5 x 128 char array
- Circular buffer overwrites oldest message

Many other examples

- Mobile phone emergency handling: 911
- EPOC fixed sized string class `Tbuf<int>`
- Real time systems:
 - E.g. Nortel Smalltalk Telephone Exchange
- 80-col punched cards
- DOS 8.3, UNIX 14-char filenames
- DOS Buffers=
- Windows Notepad, Phone Dialler
- FORTRAN and COBOL

9-Mar-00 © Charles Weir, James Noble 7

Fixed Data Structure

How can you always have enough memory?

- Memory failure is unacceptable
 - in particular kinds or parts of systems

Allocate fixed structures, avoid dynamic allocation.

- Allocate enough space to store everything you need
- Keep it unused until it is needed
- Use static arrays, circular buffers, tables...

9-Mar-00 © Charles Weir, James Noble 8

Hey — this is a pattern...

- A pattern is a
 - *Solution* to a
 - *Problem* in a
 - *Context* with a list of
 - *Related Patterns*
- Every pattern has a
 - *Name*
- Also includes:
 - *Forces* the pattern resolves
 - *An Example*
 - *Consequences* of using the pattern
 - At least three *Known Uses*

9-Mar-00 © Charles Weir, James Noble 9

Fixed Allocation: Forces and Consequences

9-Mar-00 © Charles Weir, James Noble 10

Implementation

- In C++, use static allocation to create an array of objects
- In Java or Smalltalk, pre-allocate objects when the program starts, and store in Singletons, static variables, or class variables
- In C or C++, be *very* careful not to overflow the array bounds
- In any language, handle overflows
- Record the size of the structure in a CONSTANT
 - or in C++ use `sizeof()` to access it

9-Mar-00 © Charles Weir, James Noble 11


Handling Overflow

- **Fixed User Interface** — let users see just the available memory
- **Deferring Requests** — avoid reading a new object until memory is available
- **Reducing Quality** — keep just as much of an object as you can
- **Fail older requests** — trash the oldest object
- **Fuggedaboutit** — delete the new object
- **Signal an error** — dialog boxes, exceptions, often this can be handled higher up in the system, but can be lots of work,

9-Mar-00 © Charles Weir, James Noble 12

Short Exercise

- Tacky-Tek's 'ClickTing' CameraPhone
 - Cheap mobile phone plus digital camera
 - 2 Mb of RAM, 300*300 pixel color screen
 - Text UI, Serial connection
- Memory budget?
 - Core Java OS: 100K
 - Phone SW: 75K
 - Mail application: ???
 - Screen RAM: ???
- How do we make it usable despite the small memory?



9-Mar-00 © Charles Weir, James Noble 13

Overview

- Fixed Data Structure
- **Partial Failure**
- Multiple Representation
- Packed Data

9-Mar-00 © Charles Weir, James Noble 14

Partial Failure: An example

- EPOC: uses RAM for both file system and process memory
 - So users may use up all the RAM
 - So memory allocation may fail at any time
- Can't afford to have applications lose all their data whenever allocation fails.
 - So can't terminate process on alloc fail
 - And even saving to disk might need allocated memory...

9-Mar-00 © Charles Weir, James Noble 15

Therefore: Partial Failure

- Any operation that can fail, may revert to a safe state
- Failure **never** causes lost data
 - At worst, user may have to free some memory in order to continue
- Some applications (e.g. EPOC Word) support 'failure modes', with reduced performance or display.

9-Mar-00 © Charles Weir, James Noble 16

Examples

- PalmOS uses the same principle
- MS Windows Applications' degraded modes:
 - Netscape/Powerpoint using default fonts
 - Powerpoint omits images
 - Photoshop stops saving 'undo' information
- Operating systems: fail one process; system and other processes continue.
 - UNIX, MS Windows NT, etc.

9-Mar-00 © Charles Weir, James Noble 17

Partial Failure Pattern

- *Allow the system to fail to a safe state*
- Abort the task that used the memory
- Or continue in a degraded mode.
- Localise the failure as far as possible

9-Mar-00 © Charles Weir, James Noble 18

Forces and Consequences

- Can't banish heap exhaustion
- Tasks may fail independently of others
- Users don't always need perfection...

9-Mar-00 © Charles Weir, James Noble 19

Implementation: Exceptions

- Task as a single function.
- Mechanisms to clean up stack resources

9-Mar-00 © Charles Weir, James Noble 20

Implementation: Larger tasks

- Task object (common in C++)
 - Object deleted or state set to 'failed'
- Task thread (easy in Java)
 - Exception terminates the thread
- Failure Modes
 - Part of the state of a large task object
 - e.g. Telephone call 'failed' state, word processor 'can't display' mode.
 - May be transient

9-Mar-00 © Charles Weir, James Noble 21

Overview

- Fixed Data Structure
- Partial Failure
- Multiple Representation**
- Packed Data

9-Mar-00 © Charles Weir, James Noble 22

Bitmaps

- Consider an object representing a bitmap read from a file
- How do we store it?
 - as a bitmap ready to "bitblt" to the screen?
 - as a compressed version?
 - as a reference to its representation in a file?
- Any of these may be correct, depending on how short of memory we are.


9-Mar-00 © Charles Weir, James Noble 23

Bitmaps Resolved

9-Mar-00 © Charles Weir, James Noble 24

More examples

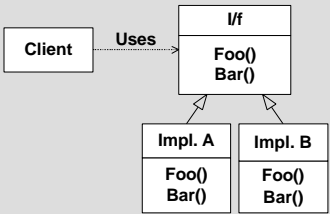
- Psion Series 5 Word Editor
- Rolfe and Nolan's Lighthouse: The Deal class
- Early IBM Smalltalk collection classes



9-Mar-00 © Charles Weir, James Noble 25

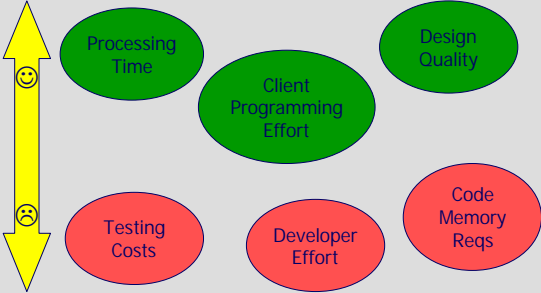
Multiple Representations

- Common abstract interface
- Clients don't worry about implementation
- E.g.



9-Mar-00 © Charles Weir, James Noble 26

Forces and Consequences



9-Mar-00 © Charles Weir, James Noble 27

Overview

- Fixed Data Structure
- Partial Failure
- Multiple Representation
- **Packed Data**

9-Mar-00 © Charles Weir, James Noble 28

The Insaniphone application


- The 'Insani-phone' telephone directory
- Stores a local Telephone Directory
 - Names and numbers
 - 200,000 subscribers



9-Mar-00 © Charles Weir, James Noble 29

Insaniphone Resolution

- Pack the data:
 - 32 area codes: 5 bits
 - Seven-digit decimal number: 24 bits.
 - Store each surname once (30,000 unique names): 18 bits for pointer
 - Three initials (5 bits each): 15 bits.
- I.e Only 4 bytes each entry (800Kb)
- Storing names at 5 bits per char: 200Kb




9-Mar-00 © Charles Weir, James Noble 30

More Examples

- Bit array classes: BitSet in both Java and C++ STL
- Y2K !
- EPOC String classes packs the class identifier in the length word.

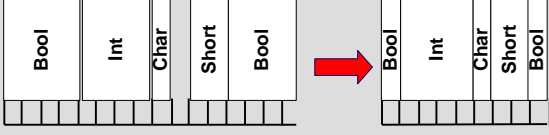

```
class TDesC8 { private:
    unsigned int iLength:28;
    unsigned int iType:4;
```
- Reuters IDN Marketstream uses bit packing and string compression to transmit Exchange prices and news.



9-Mar-00 © Charles Weir, James Noble 31

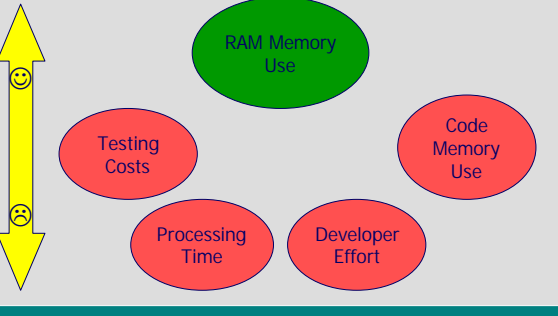
Packed Data Pattern

- Pack the data within each object
 - 'Optimise memory use' compiler flag
 - Smaller data members
 - Bytes, bitfields.
 - Arrange data structures to save padding



9-Mar-00 © Charles Weir, James Noble 32

Forces and Consequences



9-Mar-00 © Charles Weir, James Noble 33

Summary

- Several important patterns for small machines:
 - Fixed Data Structure
 - Partial Failure
 - Multiple Representation
 - Packed Data

9-Mar-00 © Charles Weir, James Noble 34

See Also

- Book: Small Memory Software
 - In Draft Form: <http://www.cix.co.uk/~cweir/>
- Summary paper
 - Proceedings of the Memory Preservation Society.

9-Mar-00 © Charles Weir, James Noble 35