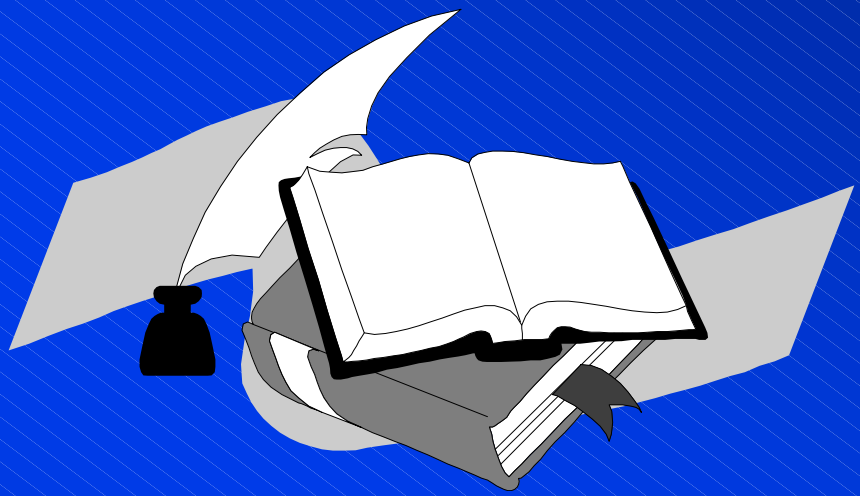
A Code Generator Writer's Diary

Jeff Froggatt



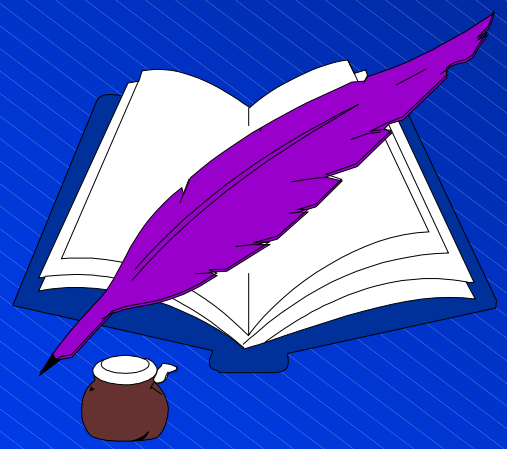
froggajn@behp72.marconicomms.com
jeff.froggatt@marconicomms.com
jfroggatt@lineone.net
http://website.lineone.net/~jfroggatt

Shlaer-Mellor User Group Conference

MARCONI COMMUNICATIONS

September 1999

Contents

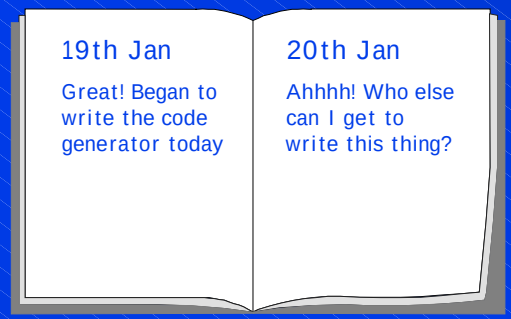


- The Diary
- The Advantages
- The Problems
- Lessons Learned
- Is Code Generation for You ?
- Beginners Guidelines
- The Ten Golden Rules

MARCONI COMMUNICATIONS

The Diary

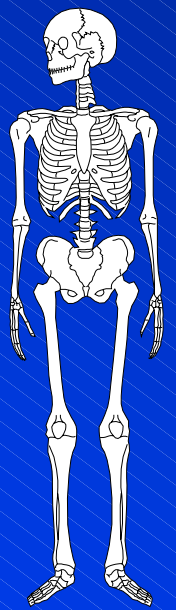
- Skeletons in the closet
- All Change
- Compilations and Greatest Hits
- The Last Drops
- Stray Dogs
- Not Just Code



MARCONI COMMUNICATIONS

The Diary

- **Skeletons in the closet**
  - Started simply with a code generator that produces skeleton code for designers to fill in
  - Existing CASE tool used for OOA modelling
  - The architecture and implementation domains used existing and well known technology
  - Much success gained with very little effort
  - Gained confidence to progress further with a bespoke architecture and implementation technologies



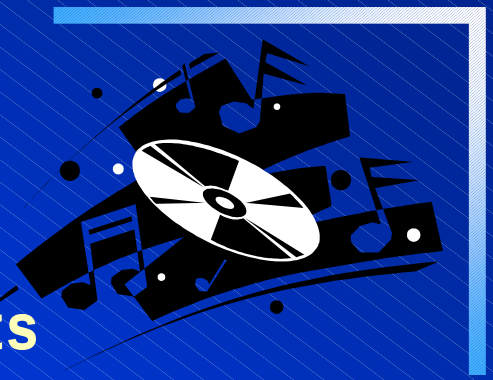
MARCONI COMMUNICATIONS

The Diary

- **All Change**
  - Started on an Objectstore based architecture
  - Started to use I-OOA
  - Oh dear, lets go off and do a completely different task
  - Really starting to think about the limitations and restrictions necessary to allow us to complete this task
  - A set of “Framework Policies” drawn up - See “Islands of OOA” SMUG 1998
  - Most of the non state action code is now generated and mapping rules written for ASL
  - Architecture changes are continually being made to accommodate new thinking

MARCONI COMMUNICATIONS

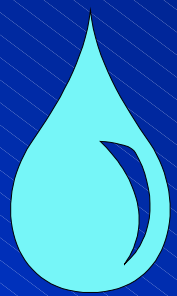
The Diary



- **Compilations and Greatest Hits**
  - Time to develop an ASL compiler !
  - Began by looking for the constructs that would give a high hit rate, left out others for the time being
  - Ensured that the compiled code was still configurable
  - Many failures were detected when the compiler is used by a second team
  - Success of hit rate depended on the teams developing the OOA, each team had an individual “dialect”
  - Almost 100% code generation of ASL but still needs manual code for bridges and complex finds

MARCONI
COMMUNICATIONS

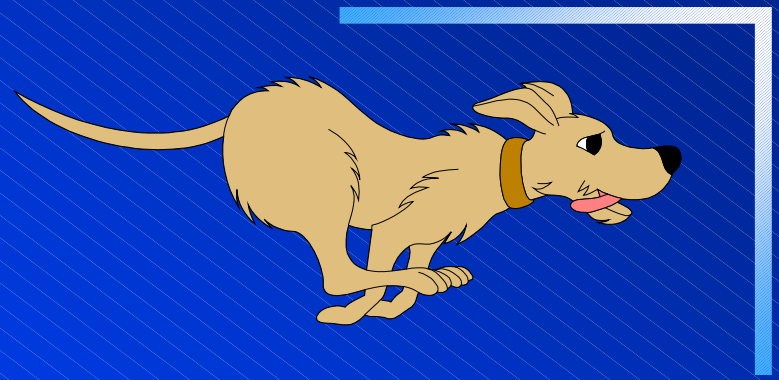
The Diary



- **The Last Drops**
  - 90% of code coverage in 10% of the development time - remaining 10% takes 90% …. or more !
  - Really started to question if the last drops were worth the effort involved
  - Started to notice limitations of code translation w.r.t optimisations required. Tagging overload !
  - Left out three categories of code generation
    - could be achieved by other OOA constructs
    - would be better optimised by hand
    - generation would require processing of more than one domain

MARCONI COMMUNICATIONS

The Diary

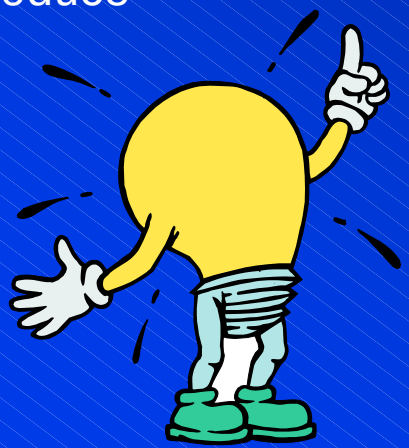


- **Stray Dogs**
  - Released the code generator to another part of the company where it took on a life of its own
  - It was converted to GNU C++ and provided support for transient instances - See “OOA Code Generation Adaptation: Re-use in practice” by Hamish Blair SMUG 1997
  - Was later modified for by yet another group to produce C source code for another architecture

MARCONI COMMUNICATIONS

The Diary

- **Not Just Code**
  - Target code is not an end in itself for a configurable code generator
  - The most recent modifications were to produce
    - test/debug code
    - model statistics
    - HTML documentation



MARCONI COMMUNICATIONS

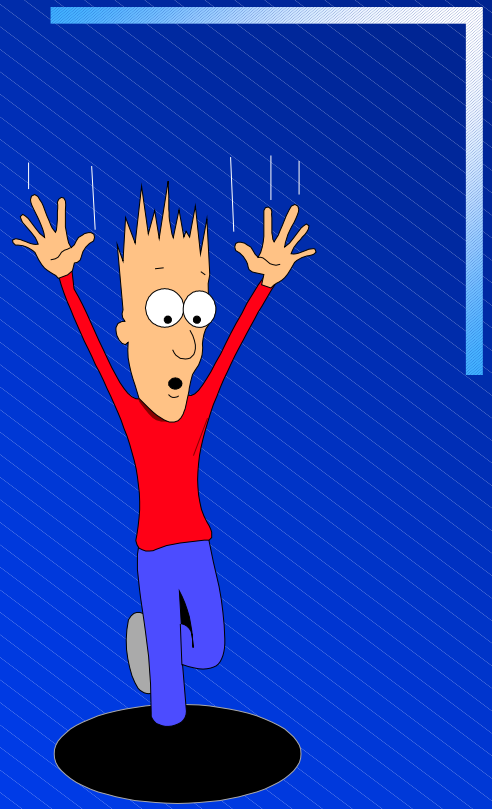
The Advantages

- Repeatability
- Future proofing
- Round and round the loop
- Self documentation
- Skills in one place

MARCONI COMMUNICATIONS

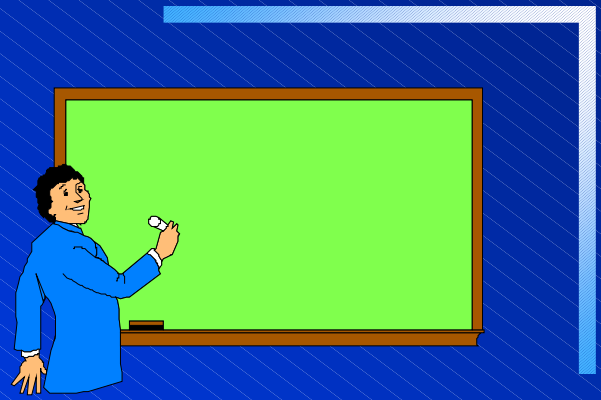
The Problems

- Performance
- Lead time for problem fixing
- Modelling constraints
- Flexibility of code



MARCONI
COMMUNICATIONS

Lessons Learned

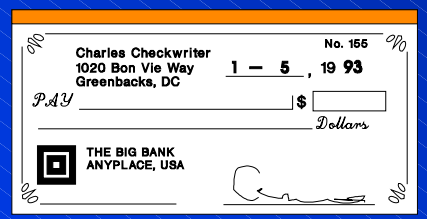


- Keep it simple
- Go all the way !
- Don't underestimate the impact of the chosen architecture and implementation techniques
- Not all OOA is the same - beware of "dialects"
- Test, test and test some more !

MARCONI COMMUNICATIONS

Is Code Generation for You ?

- Beware of the initial investment
  - better for larger projects
- It's reproducible
  - good for critical applications
- Performance can be a problem
  - don't use for high throughput parts of the application
- Leaves designers one step removed from code
  - do your staff want to do coding?



MARCONI COMMUNICATIONS

Beginners Guidelines

- Get into the architecture design early
- Start simply with skeleton code and work up
- Add ASL compiler last
- Prototype implementation technologies
- Keep code generator flexible
- Have a go !

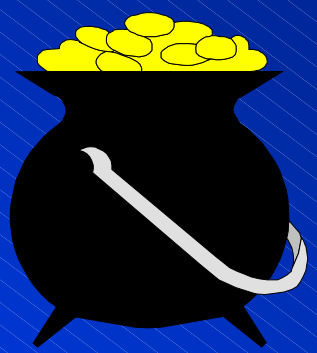
MARCONI COMMUNICATIONS

The Ten Golden Rules

- Keep it simple
- Bring the technologies up to OOA level, don't try to code generate everything
- Build in configurability
- Iterate the development
- Know your OOA of OOA

MARCONI
COMMUNICATIONS

The Ten Golden Rules

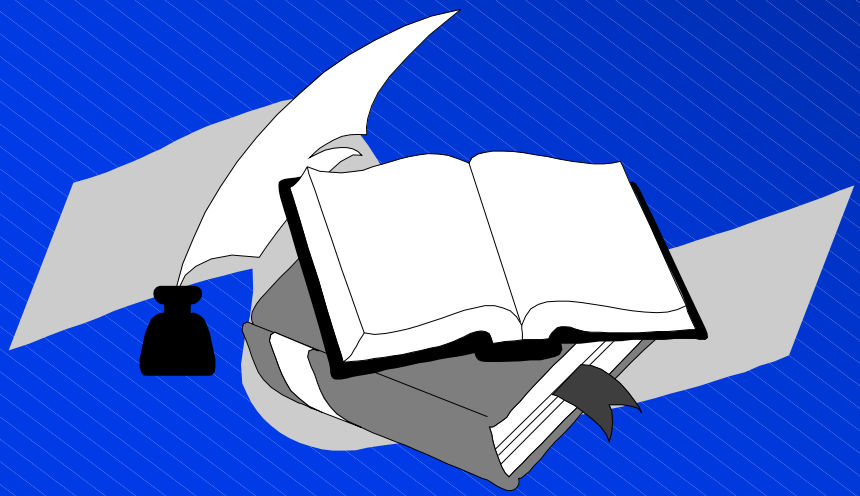


- Test all OOA “dialects”
- No special cases !
- One fact at one visit, don’t keep looking up the same information
- Know your ASL .... and be patient with it
- Don’t do it ! Buy I-CCG

MARCONI COMMUNICATIONS

A Code Generator Writer's Diary

Jeff Froggatt



froggajn@behp72.marconicomms.com
jeff.froggatt@marconicomms.com
jfroggatt@lineone.net
http://website.lineone.net/~jfroggatt

Shlaer-Mellor User Group Conference

MARCONI COMMUNICATIONS

September 1999