Code Generation Supported by a Pattern-based Design Methodology

Chisanga Mwelwa and Dr Michael J Pont
Embedded Systems Laboratory
University of Leicester

Dr David Ward
MIRA Ltd

Embedded Systems Show
Birmingham, UK, 13th -14th October 2004

Embedded Systems Laboratory
http://www.le.ac.uk/eg/embedded

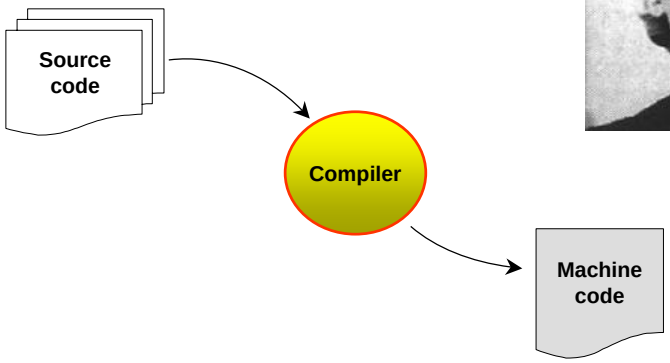
Overview

- I will argue that design patterns provide a reliable means of supporting automated code generation in embedded systems development
- I will begin by giving an overview of design patterns
- I will then consider the challenges involved in pattern-based code generation
- I will describe a prototype tool for pattern-based code generation

Embedded Systems Laboratory
http://www.le.ac.uk/eg/embedded

Compilers: the first code generators

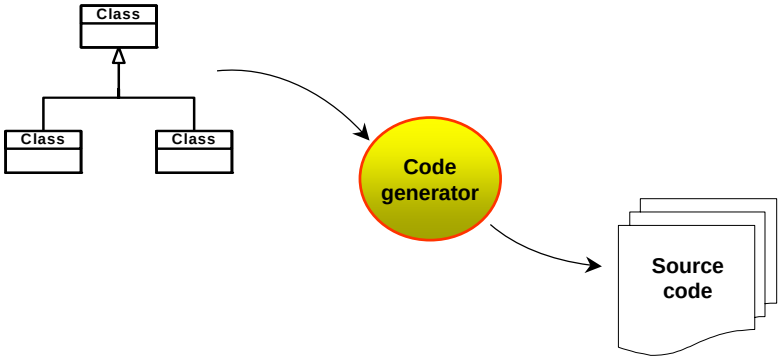
- 1951 - 1952 Admiral Grace Murray Hopper develops A-0, the first compiler



Embedded Systems Laboratory
http://www.le.ac.uk/eg/embedded

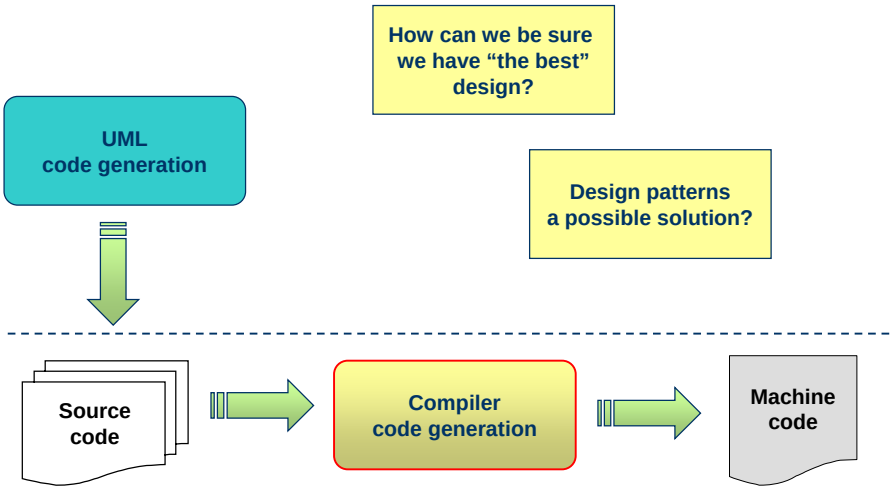
Model-based code generators

- Code generation has evolved, it is now also based on software design models such as UML



Embedded Systems Laboratory
http://www.le.ac.uk/eg/embedded

Code generation: the state-of-the-art



Embedded Systems Laboratory
http://www.le.ac.uk/eg/embedded

Design patterns

- The concept emerged from the work of the architect - Christopher Alexander:

  "Each pattern describes a recurring problem and then describes the core of the solution to that problem in such a way ***that you can use this solution a million times over****."*

- In a "A Pattern Language", Alexander and colleagues proposed 253 coherent design patterns for designing and building homes, cities etc.

Embedded Systems Laboratory
http://www.le.ac.uk/eg/embedded

Alexander's patterns for a porch design

- Sunny Place
- Front Door Bench
- Raised Flowers
- Columns at the Corners

Other patterns

- Over the years Alexander's concept has been applied in other fields…
- Organisational patterns e.g. giving presentations
  Peter Ruthven-Stuart (2001)
- Patterns for telecommunications systems
  Linda Rising (2001)
- Object-oriented software patterns
  Gamma et al (1995)

Example: HANDS IN VIEW

- Problem: Skier fails to commit downhill on steeps and bumps, resulting in slides, backward falls, and "yard sales."

- Solution: Concentrate on keeping the hands in view. Bring them into sight immediately after each pole plant and turn

AUTHOR:

ANONYMOUS SKI INSTRUCTOR
SOMEWHERE IN UTAH, USA

9

Embedded Systems Laboratory
http://www.ie.ac.uk/reg/embedded

Software design patterns

- In 1995 Erich Gamma and colleagues published a set of reusable object-oriented design patterns
- Resulted in the creation of a worldwide "pattern community" where pattern enthusiasts discuss "everything patterns"
- PLoP (Pattern Languages of Programming) conferences include: EuroPLoP, VikingPLoP, ChilliPLoP, KoalaPLOP, MensorePLoP and SugarLoafPLoP

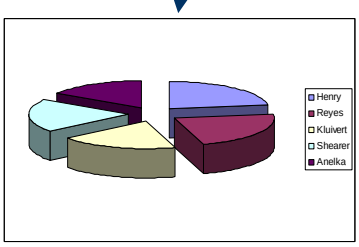
10

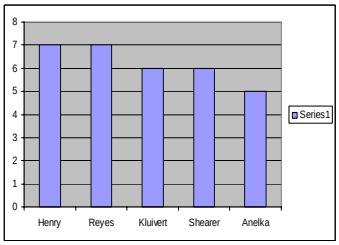
Embedded Systems Laboratory
http://www.ie.ac.uk/reg/embedded

Example: OBSERVER

- Problem: How can you keep multiple views of a single object in synchronous with the object?

Top Scorers	
Henry	7
Reyes	7
Kluivert	6
Shearer	6
Anelka	5





Embedded Systems Laboratory
http://www.le.ac.uk/eg/embedded

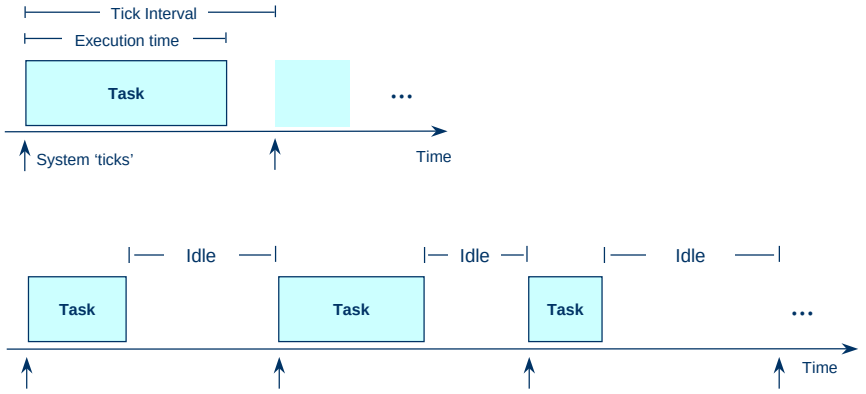
Pattern language for embedded systems

- Over the last 8 years a pattern language has evolved in the Embedded Systems Laboratory
- Patterns for Time-Triggered Embedded Systems – "PTTES collection"
- At present there are over 70 patterns in the collection e.g.:
  - Monitoring and control patterns
  - User interface patterns e.g. LCD PANEL and PC LINK (RS232 interface)

For more patterns including example source code visit:
http://www.le.ac.uk/engineering/mjp9/patterns.html

Embedded Systems Laboratory
http://www.le.ac.uk/eg/embedded

Example: Co-operative Scheduler



13

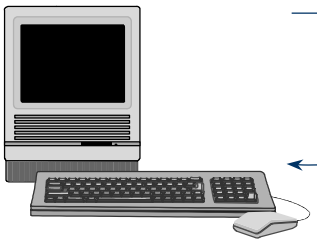
Embedded Systems Laboratory
http://www.le.ac.uk/eg/embedded

Example: Multi-Stage Task

Current core temperature is 36.678 degrees

All characters written immediately to buffer (very fast operation)

Buffer



Scheduler sends one character to PC every 10 ms (for example)

14

Embedded Systems Laboratory
http://www.le.ac.uk/eg/embedded

Current application of patterns

Pattern catalogue
(Most come with code examples)

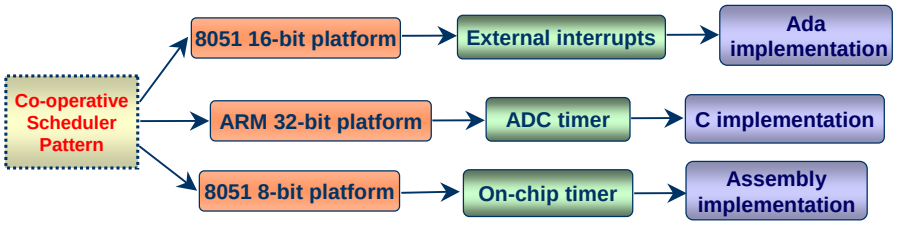
Developers

Source Code

Pattern-based code generation

- A good pattern is expected to have at least three known uses/implementations i.e. one pattern can be implemented in more than one way
- This one-to-many relationship of patterns challenges those attempting to use patterns for automated code generation

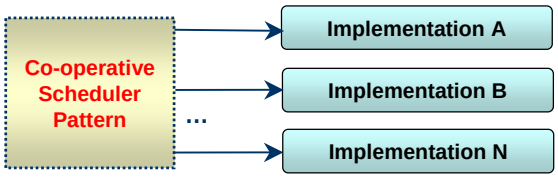
Implementation of CO-OPERATIVE SCHEDULER



Embedded Systems Laboratory
http://www.le.ac.uk/eg/embedded

Pattern-based code generation

- Developing a tool that can implement all the possible implementations of a pattern is close to impossible



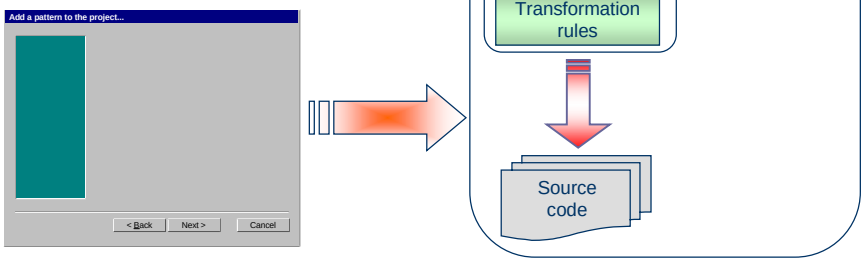
Embedded Systems Laboratory
http://www.le.ac.uk/eg/embedded

Our approach – patterns as components

- Our approach is to develop a tool that can generate code from a subset of the possible implementations of a defined collection of patterns
- The result is a tool that uses patterns to support the implementation of - and integration of - a set of software components

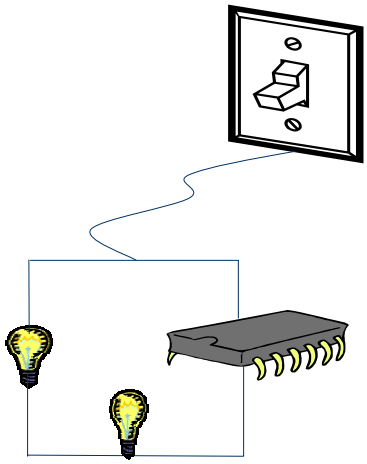
Architecture of a pattern-based tool

- User has control of tool via a “wizard”

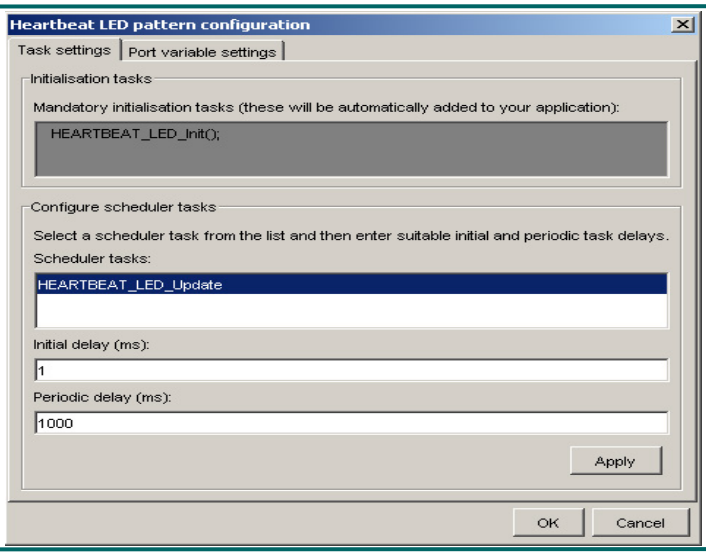


Tool application (simple example)

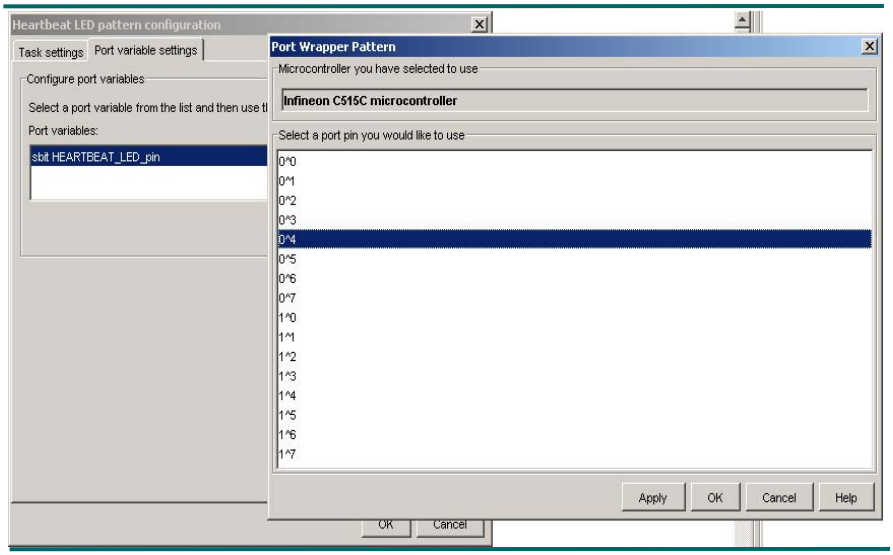
- System functions:
  - One flashing LED
  - A second LED controlled by a switch
- Six patterns applied:
  - Extended 8051
  - Crystal Oscillator
  - Co-operative Scheduler
  - Port Wrapper
  - Switch Interface
  - Heartbeat LED



Pattern implementation

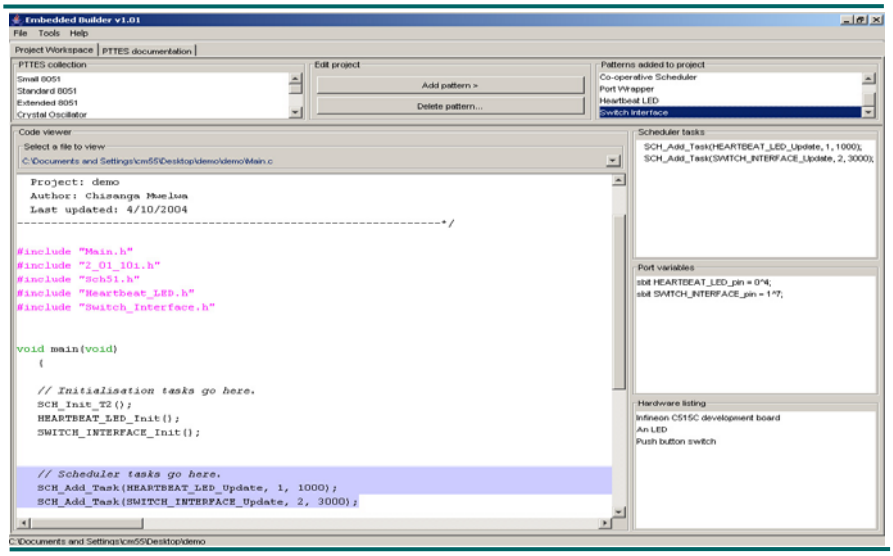


Configuration of microcontroller hardware



Embedded Systems Laboratory
http://www.le.ac.uk/eg/embedded

Generated code



Embedded Systems Laboratory
http://www.le.ac.uk/eg/embedded

Current tool capabilities

- At present tool provides an efficient means of integrating design patterns into simple reliable embedded systems
- 800 LOC integrated into a system in 8 minutes
- Code compiled by Keil compiler and generated no errors
- Supports software maintenance – adding and removing of a systems software components in the form of patterns

SIMPLE EXAMPLE

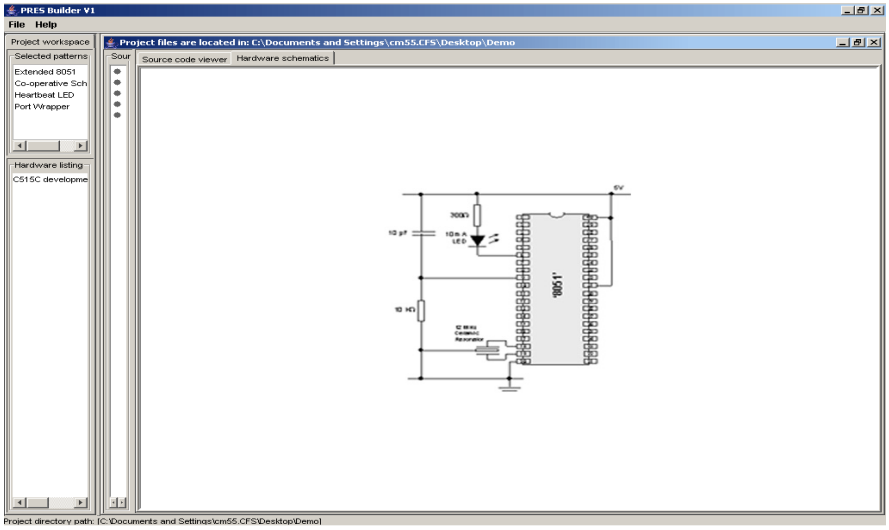
– BUT –

INDICATIVE OF THE POTENTIAL OF A PATTERN BASED CODE GENERATOR

Current work: hardware support

- Our patterns also support hardware development – this feature can be included in the tool
- SystemC is currently being considered for the hardware implementation
- SystemC is a C++ open source library used for hardware design, validation and simulation

Current work: hardware support



Embedded Systems Laboratory
http://www.le.ac.uk/eg/embedded

Conclusions

- Complete tool is expected to provide an efficient and reliable means of developing embedded systems by using design patterns
- Using design patterns, both hardware and software can be developed simultaneously in one environment
- A pattern-based tool has positive implications for software maintenance (adding and removing software components)

Embedded Systems Laboratory
http://www.le.ac.uk/eg/embedded

References

- C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fisksdahl-King and S. Angel, "A Pattern Language", Oxford University Press, New York, 1977.
- F. Budinsky, M. Finnie, J. Vlissides and P. Yu, "Automatic Code Generation from Design Patterns", IBM Systems, Vol. 35, (2), Pp. 151-171, 1996.
- E. Gamma, R. Helm, R. Johnson and J. Vlissides, "Design Patterns: Elements of Reusable Object-Oriented Software", Addison-Wesley, 1995.
- C. Mwelwa, M.J. Pont and D Ward (2004) "Using Patterns to Support the Development and Maintenance of Software for Reliable Embedded Systems: A Case Study", Proceedings of the IEE / ACM Postgraduate Seminar on "System-On-Chip Design, Test and Technology", Loughborough, UK, 15 September 2004, Published by IEE. ISBN: 0 86341 460 5 (ISSN: 0537-9989).
- C. Mwelwa and M.J Pont (2003) "Two Simple Patterns to Support the Development of Reliable Embedded Systems", Proceedings of the 2nd Nordic Conference on Pattern Languages of Programs (VikingPLoP'03), September 19-21, 2003, Bergen, Norway.
- M. J. Pont, "Patterns for Time-Triggered Embedded Systems", Addison-Wesley, 2001.
- L. Rising, "Design Patterns in Communications Software", Oxford University Press, 2001.
- J. E. Sammet, "History of IBM's Technical Contributions to High Level Programming Languages", IBM Journal of Research and Development, Vol. 25, (5), Pp. 520-534, 1981.