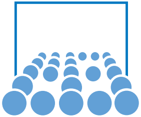
UML, Code generation and Software libraries

Advanced Programming Tutorials

Erik Wannerberg
Supervisors: Roland Wittmann, Philipp Neumann

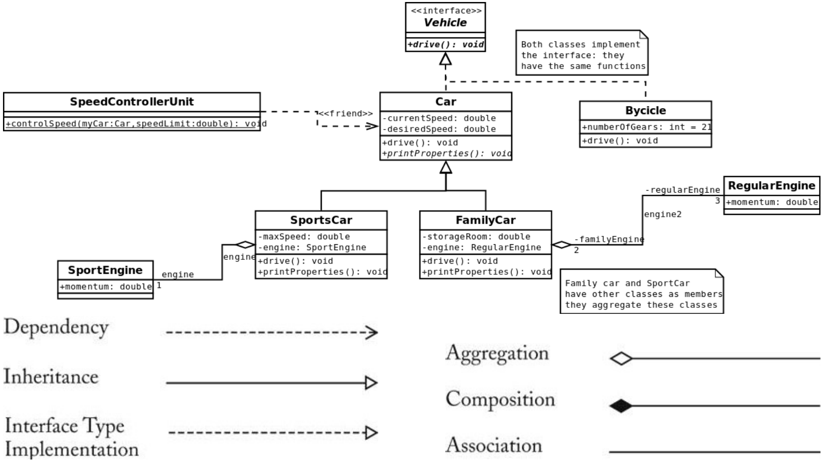
December 8, 2015



Content

1. **Overview**
2. **A brief introduction to UML**
3. **Code generation - what? why?**
   - 3.1 dia2code
   - 3.2 Matlab Coder
4. **Libraries? But I don't want to read books...**
   - 4.1 bullet
   - 4.2 SDL
   - 4.3 boost libraries
   - 4.4 Linear algebra
   - 4.5 VTK
   - 4.6 MPI
   - 4.7 How to use?

UML Class diagram - Example



<<interface>>
Vehicle
+drive(): void

Both classes implement the interface: they have the same functions

SpeedControllerUnit
+controlSpeed(myCar:Car,speedLimit:double): void

<<friend>>

Car
-currentSpeed: double
-desiredSpeed: double
+drive(): void
+printProperties(): void

Bycicle
+numberOfGears: int = 21
+drive(): void

RegularEngine
+momentum: double

-regularEngine
3
engine2

SportsCar
-maxSpeed: double
-engine: SportEngine
+drive(): void
+printProperties(): void

FamilyCar
-storageRoom: double
-engine: RegularEngine
+drive(): void
+printProperties(): void

-familyEngine
2

SportEngine
+momentum: double

engine
engine
1

Family car and SportCar have other classes as members they aggregate these classes

UML - Unified Modeling Language

"*The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering, that is intended to provide a standard way to visualize the design of a system.*" (Wikipedia).

Widely known and used - *very* many different types
Structural diagrams:

- **Class diagram**
- Package diagram
- ...

Behavioral diagrams:

- Activity diagram
- **State diagram**
- Communication diagram
- ...

Tutorial: http://www.sparxsystems.com/resources/uml2_tutorial/uml2_classdiagram.html

Code generation - what? why?

Generally: the creation of code from other code

Lower-level, more explicit, verbose output from higher-level

- Macros (C preprocessor `# define A(arg) arg B = 1337;`)
- Templates - small, big
- Visual programming (visually based IDEs)
- Transcompilers

But also:

- Compilers (old meaning!) and disassemblers

(See `http://www.methodsandtools.com/archive/archive.php?id=86` for a nice historical outlook)

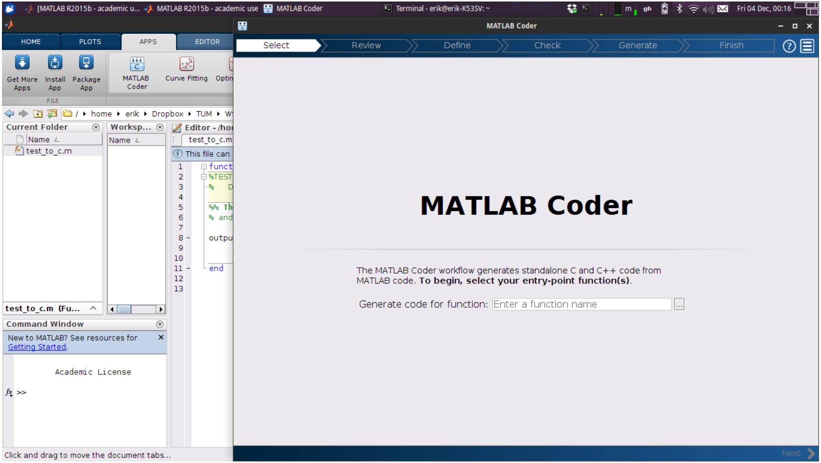
dia2code

Code generation using diagrams drawn in `dia`

- Generation of code stubs from UML class diagrams
- Several languages:
  - C, C++, C#
  - Java, python, ruby
  - PHP, PHP5, SQL
  - ...

Matlab Coder

Generation of C/C++ code from Matlab functions



The MATLAB Coder workflow generates standalone C and C++ code from MATLAB code. **To begin, select your entry-point function(s).**

Generate code for function: Enter a function name

Libraries? But I don’t want to read books...

Why invent the wheel twice? (When someone else has invented a much rounder, lighter, more stable one for half the price)

Allows you to import large works of code, (typically) heavily optimised...

- for performance
- for stability (bug/error -free)
- for usability (sometimes)
- ...and contains tools you couldn’t even dream about

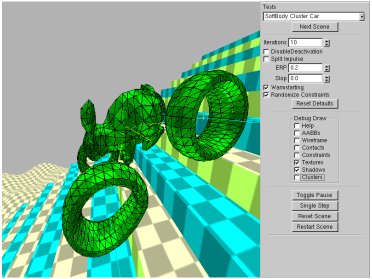
bullet

Physics simulation library

Used in many games, movies software; e.g.:

- Games: Grand Theft Auto IV, Grand Theft Auto V, ...
- Movies: 2012, Hancock, Sherlock Holmes, Shrek 4
- Blender Game Engine (free 3D production suite)

http://bulletphysics.org/



Tests: SoftBody Cluster Car

Iterations: 10

□ DisableDeactivation

□ Split Impulse

ERP: 0.2

Slop: 0.0

☑ Warmstarting

☑ Randomize Constraints

Debug Draw

□ Help

□ AABBs

□ Wireframe

□ Contacts

□ Constraints

☑ Textures

☑ Shadows

□ Clusters

Simple DirectMedia Library

Cross-platform library for providing access to I/O devices such as keyboard, joystick, mouse, but also graphics through OpenGL and Direct3D

Bindings for many languages (C, C++, Pascal, Python, C#...)

Used in games such as:

- Angry Birds
- Unreal Tournament
- Many games provided by Steam and Humble Bundle...

http://www.libsdl.org/

boost libraries

Peer-reviewed set of libraries designed for many tasks

Examples:

- Linear algebra
- Pseudorandom number generation
- Multithreading image processing
- Regular expressions
- Unit testing

Several implemented into the C++11 standard! (Even more in C++17)
http://www.boost.org/

Linear algebra

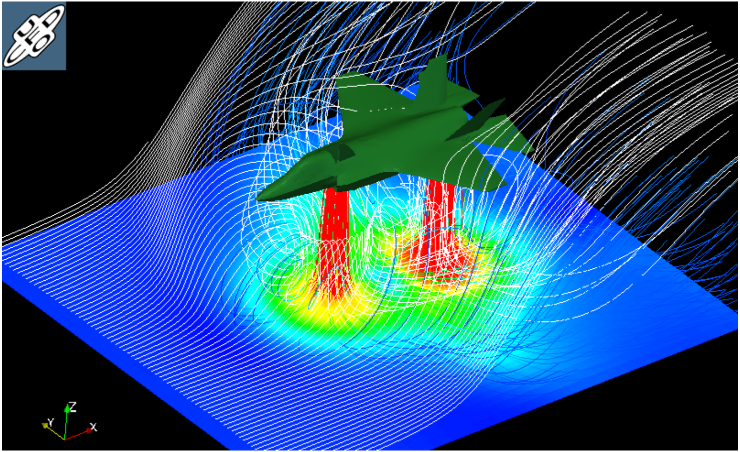
$$\begin{bmatrix} L & A & P & A & C & K \\ L & -A & P & -A & C & -K \\ L & A & P & A & -C & -K \\ L & -A & P & -A & -C & K \\ L & A & -P & -A & C & K \\ L & -A & -P & A & C & -K \end{bmatrix}$$

BLAS - Basic Linear Algebra Subroutines

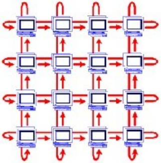
- The subroutines called by basically every implementation with linear algebra
- Architecture-specific implementations optimised for performance on *your* computer
- Included in more user-friendly libraries **LAPACK**, scalable counterpart **ScaLAPACK** etc.

VTK- the Vizualisation ToolKit

Huge visualisation library with bindings for C++, Python... which also provides the back-end for Paraview



MPI- Message Passing Interface



Library for parallel computing, used in most supercomputers (for example SuperMUC)

How to use?

- Need to include source files - compiler can’t do magic with `<header.h>`!
    - Solution: `-I<source_dir>`
- Need to link to libraries!
    - Can link just like object files
    - `-l<library_name>`
    - `-L<library_dir>`

Valgrind

Tool for finding memory leaks. Can also find out-of-bounds accesses and usage of uninitialised values.

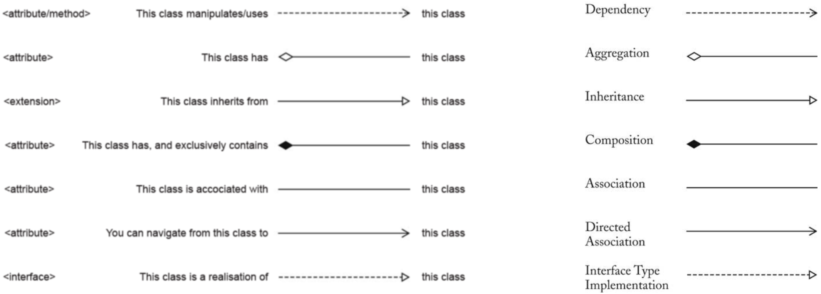
Tutorial: http://www.cprogramming.com/debugging/valgrind.html

Install:
Linux package available (`sudo apt-get install valgrind`), otherwise have to build yourself! (instructions available)

Run:
Like `gdb`: run like `valgrind [options] <program> <program_arguments>`

UML Class diagram

Ways of describing relations between classes. The lines are:


<attribute/method> This class manipulates/uses this class
<attribute> This class has this class
<extension> This class inherits from this class
<attribute> This class has, and exclusively contains this class
<attribute> This class is accociated with this class
<attribute> You can navigate from this class to this class
<interface> This class is a realisation of this class
Dependency
Aggregation
Inheritance
Composition
Association
Directed Association
Interface Type Implementation

Source: http://wiki.msvincognito.nl/Study/Bachelor/Year_2/Object_Oriented_Modelling/Summary/Object-Oriented_Design_Process
(also has more detailed descriptions)