OGEL:
LEGO Mindstorm Controller Language

Michael Ching

Gerardo Flores

Matt Kalish

Charles O'Donnell

May 13, 2003

LEGO Mindstorms

- Robotic architecture that allows one to build components from Legos and control sensors and motors from a microcontroller
- Complexity of microcontroller can be chosen (Scout, RCX 1.0, RCX 2.0, CyberMaster)
- Different development environments may be chosen that vary from a drag and drop approach to a C-syntax based language

Not Quite C

- Targets experienced programmers
  - C syntax and semantics
  - Textual language removes the graphical metaphors that limit the type of programs that may be written
  - Supported by several IDEs (WinNQC, MacNQC, Visual NQC, NQCEdit, Bricx Command Center)
  - Not modular in nature
  - Does not support hierarchy or abstraction of code
  - Questionable event-based support

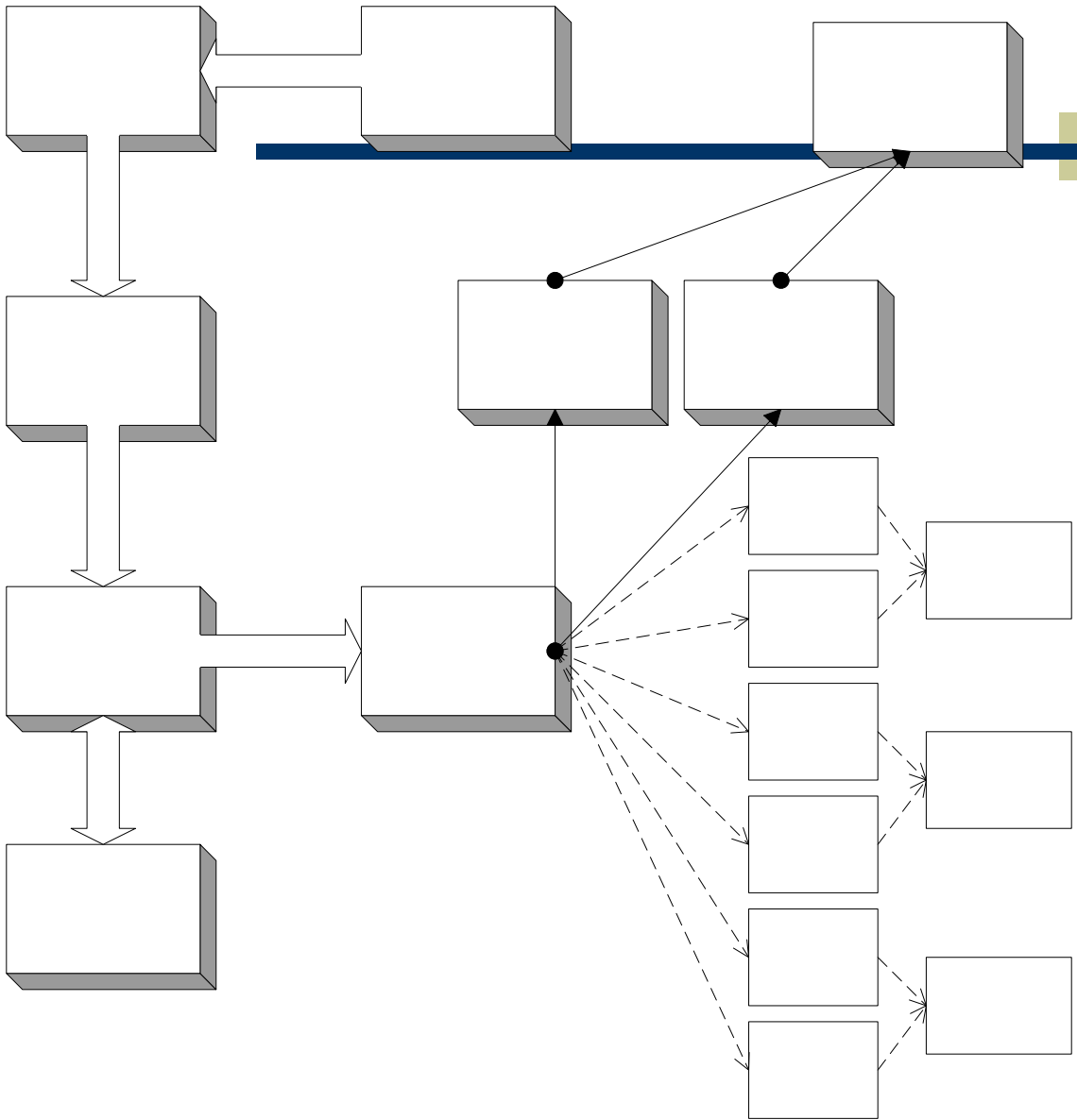
OGEL

- Designed to support the RCX 1.0 implementation but offers support for future firmware (Compatible with 2.0)
- Merge desirable characteristics of available technologies
- Built scalable hierarchy which improves code reuse

OGEL (Cont'd)

- Able to create code for multiple target microcontrollers from one file
- Improved message passing architecture
- Improved modularity which leads to a more stable hierarchy
- Allows imbedded self-contained Not Quite C code
- Permits separation of RCX, task, message, and function declarations into library files which can be added to any existing code by means of include statements

Architecture



Preprocesso

Architecture Discussion

- Common compiler components: Lexar/Parser, Semantic Checking, IR Translation, Translation to NQC
- Back-End

Sample OGEL Code

```
rcx bottom_rcx {
        A = fwd_wheel;
        B = arm;
        1 = front_hit;
        2 = light;
}

message hit;

task main() {
}

task hit_enemy(in front_hit, out arm, out fwd_wheel) {
}

void arm_swing(out arm) {
}
```

Watch This Puppy run!

Division of Work:

Front-End: Michael Ching / Gerardo Flores

Back-End: Charles O’Donnell

Docs, Research, LEGOs: Matt Kalish