Project Presentation

"Generic Python API for Robot Control"

Group 6
Thomas Nourse
Greg Hoch
December 12, 2005

Agenda

- **Project Overview**
  - Problem
  - Goals
  - Product
  - Demo
- **Results and Lessons**
  - Results
  - Changes
  - Evaluation
  - Lessons Learned

Project Overview
The Problem

- Lack of code portability between robotic platforms.
  - Platforms have different interfaces to use the same hardware.
  - Platforms have different hardware with similar capabilities.
    - Range Finders (Laser vs. Sonar)
    - Drive Systems (Differential vs. Holonomic)
    - Communication (Serial vs. WiFi)

Project Overview
Goals

- "Create a high level Python API which will run the same robot behaviors on multiple robots."
- Portability
- Extensibility
- Simplicity

Project Overview
Goals - Robotic Landscape

Project Overview
Product

- Designed and implemented both the API and an example implementation on the Nomad Scout.
  - API : Hides the robotic platform from user if he/she desires.
  - Implementation: Bridges the gap between the API and the individual platform implementation
- Documented with Doxygen the API and process necessary for extension.

Project Overview

Product – API

- Divided into sensors, actuators and configuration.
- Single interface (RobotAPI) provides access to all robotic functionality.
  - Initializes from an easily interchangeable init file.
- Programmatic way to discover robot type and general sensor / actuator capabilities.

Project Overview

Product – Implementation

- Nomad scout as our test platform.
  - Useful for testing API conceptually.
  - Integral in documentation by example.
- Implemented serial interface for basic communication with the robot.
- Connected this interface with the API.
- Functional test using a virtual null modem cable testing output against traces from the original C interface.

Project Overview
Product – Built in Extensibility

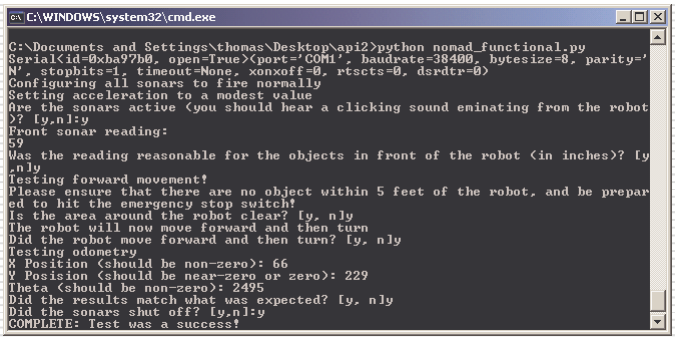
- Designed specifically to add new robots and robot functionality.
  - Robots split among directories / packages, easily interchangeable.
  - Extensible methods for determining the current type of robot being used.
- Designed with flexibility to ease addition of new layers such as:
  - GUI Debugging Interface
  - Teleoperation Interface

Project Overview
Demo – Basic Movement

Command: "python nomad_functional.py"

Project Overview

Demo – Basic Movement



An example run of a qualitative functional test.

Project Overview

Demo – Controlled Movement

```
while api.getRangeSensors()[16].getValue(0) > 12:
    time.sleep(1)
    api.getVelocityActuators()[0].setVelocity(50,50)
api.getVelocityActuators()[0].setVelocity(0,0)
```

Project Overview

Demo – Dancing!

```
for i in xrange(3):
self._rap.getVelocityActuators()[0].setVelocity(100,-100) == 1
            time.sleep(3)
self._rap.getVelocityActuators()[0].setVelocity(-100,100) == 1
            time.sleep(3)
self._rap.getVelocityActuators()[0].setVelocity(150,-150) == 1
            time.sleep(3)
self._rap.getVelocityActuators()[0].setVelocity(-150,150) == 1
            time.sleep(3)
self._rap.getVelocityActuators()[0].setVelocity(-200,200) == 1
            time.sleep(3)
self._rap.getVelocityActuators()[0].setVelocity(200,-200) == 1
            time.sleep(3)
self._rap.getVelocityActuators()[0].setVelocity(250,-250) == 1
            time.sleep(3)
self._rap.getVelocityActuators()[0].setVelocity(-250,250) == 1
            time.sleep(3)
```

Results and Lessons

Results – Orig. Picture of Success

- Follow XP and course practices
  - Meet all deliverable expectations
  - Pair programming
- Significant behavior on AIBO and ER1
- Client should use it within a year
- One additional device and one additional robot added within a year
- Client satisfied

Results and Lessons

Results – Pict. of Success Changes

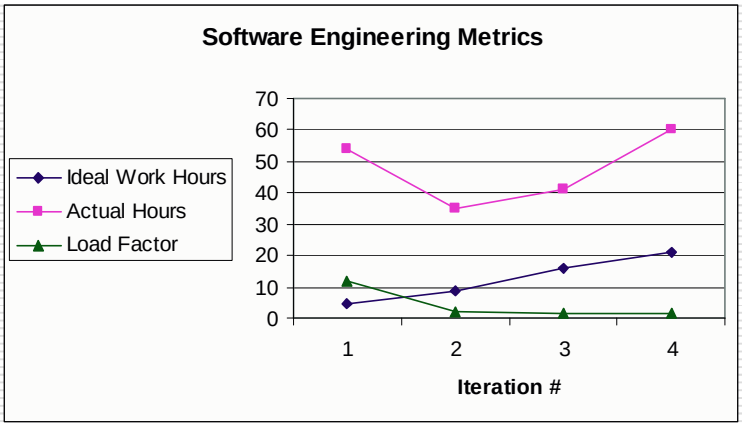
- Project focus changed from the AIBO to the Nomad Scout
  - Complexity of the AIBO
  - Group familiarity with the Nomad Scout
- Client's focus changed from the AIBO to the ER1

Results and Lessons

Results – Success?

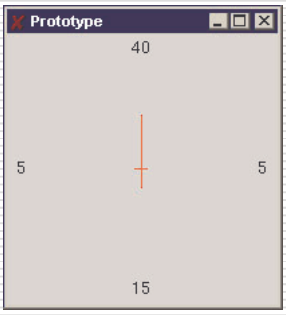
- XP and course practices? YES
- Significant behavior on Nomad? YES
- Client use within a year? TBD
- Additional device/robot? TBD
- Client satisfied? TBD
- Get an A in this course? WE HOPE SO

Results and Lessons
Results – Metrics



Results and Lessons
Changes – Prototype Results

- Provided early image of the project
- Redirected project
  - Platform AIBO -> Nomad Scouts
  - Actuator API -> Sensor API specifically Range Finder



Results and Lessons

Changes – Low Time Estimation

- Experience in Computer Networks gave us better skill in estimating time requirements for network programming.
- Determined we had underestimated our time needed for teleoperation and network communication.
- Dropped priority of involved stories as a result.

Results and Lessons

Changes – Team Member Dropped

- Left group with two members.
- Caused drop in available time resulting in lower prioritization of more requirements.
  - Debugging framework
  - Logging driver
  - Information GUI
- Focused on achieving picture of success when slimming down project.

Results and Lessons
Strengths

- Loss of a Partner
  - Required replanning the project to bring it within scope of a two person group
- Consistently meeting deadlines
- Keeping client informed
  - Weekly status meetings
  - Website

Results and Lessons
Strengths

- We have been partners on previous projects
  - Work well together
  - Easy to recognize and capitalize on our strengths as developers
- Centralized Project Wiki
  - Kept team members and clients universally updated
  - Acted as a repository for our course documents
  - Kept meeting minutes in case someone was absent or we needed to refer to what was said
  - Common source for resource material

Results and Lessons
Weaknesses

- We have been partners on previous projects
  - Easy to fall into old habits, which did not fit within the XP framework for development.
- Time
  - Difficult to consistently find common time to work together.

Results and Lessons
Weaknesses

- Pair Programming
  - Did not schedule time for pair programming
  - Hard to find overlapping free time that was not taken up with class assignments
- Stories
  - Difficulty understanding scope of stories
  - Difficulty forming comprehensive stories for project

Results and Lessons
Lessons Learned

- Nomad Serial Interface
  - Written without originally testing on the robot
  - Unit and functional tests simulated communication and checked bounds
  - Used traces from C-API communications
  - Worked the first time!
- Lesson: XP Works!

Results and Lessons
Lessons Learned

- Regular review of results and revision of requirements
  - Prototype – new direction
  - Learning more information – revise story priority
  - Loosing a partner – dropped stories
- Consistent meetings
- Robots are hard!

Conclusions