Behaviour patterns for a sensor platform

Manuel Meder
July 26, 2011
Content

1. Review: Where did we start from?
2. Review: Pleo’s software architecture.
Content

1. Review: Where did we start from?
2. Review: Pleo’s software architecture.
3. Two new behaviours for Pleo.
4. The main task of this thesis.
Content

1. Review: Where did we start from?
2. Review: Pleo’s software architecture.
3. Two new behaviours for Pleo.
4. The main task of this thesis.
5. What does/doesn’t yet work?
Content

1. Review: Where did we start from?
2. Review: Pleo’s software architecture.
3. Two new behaviours for Pleo.
4. The main task of this thesis.
5. What does/doesn’t yet work?
8. Summary.
Review: Where did we start from?

The idea:

Extend the experimental platform of the SFB/TRR 62 with an autonomous sensor platform in order to gain more possible ways the user can interact with the Companion-System that runs on the experimental platform.
Review: Where did we start from?

**The idea:**

Extend the experimental platform of the SFB/TRR 62 with an autonomous sensor platform in order to gain more possible ways the user can interact with the Companion-System that runs on the experimental platform.

### Already covered

1. Chosen the autonomous sensor platform: Pleo.
2. Considered the technical aspects of Pleo.
3. Decided for a medium to exchange data.
4. Designed a reliable and robust protocol.
Review: Pleo’s software architecture

- 3 layered architecture

- Pleo’s behaviour determined by drives

In order to analyse the potential of the scripts, two simple behaviours are created that use different functionalities.
Two new behaviours for Pleo: Pleostrument

Pleo becomes a drum set

This behaviour only uses sensor events and leaky integrators, which work against the users effort.
Two new behaviours for Pleo: Pleostrument

Pleo becomes a drum set

This behaviour only uses sensor events and leaky integrators, which work against the users effort.

Pleostrument

- Touch-Sensor events increase count.
- Leaker reduces that count again.
- Level of count determines audio speed.
- Max. level plays final sound and resets count.
- Leaky integrators work as timers in Pleo's interpreter script.
Two new behaviours for Pleo: Pleostrument

Pleo becomes a drum set

This behaviour only uses sensor events and leaky integrators, which work against the users effort.

- Touch-Sensor events increase count.
- Leaker reduces that count again.

![Diagram showing the Pleostrument behaviour pattern with sensor events and leaky integrators.](attachment:diagram.png)
Two new behaviours for Pleo: Pleostrument

Pleo becomes a drum set

This behaviour only uses sensor events and leaky integrators, which work against the users effort.

- Touch-Sensor events increase count.
- Leaker reduces that count again.
- Level of count determines audio speed.
- Max. level plays final sound and resets count.
Two new behaviours for Pleo: Pleostrument

Pleo becomes a drum set

This behaviour only uses sensor events and leaky integrators, which work against the users effort.

- Touch-Sensor events increase count.
- Leaker reduces that count again.
- Level of count determines audio speed.
- Max. level plays final sound and resets count.
- Leaky integrators work as timers in Pleo’s interpreter script.
Two new behaviours for Pleo: Pleohoo

Pleo plays noisy ghost

The use of "user properties" in combination with sensors determine what mode Pleo is in.
Two new behaviours for Pleo: **Pleohoo**

Pleo plays noisy ghost

The use of "user properties" in combination with sensors determine what mode Pleo is in.
Two new behaviours for Pleo: Pleohoo

Pleo plays noisy ghost

The use of "user properties" in combination with sensors determine what mode Pleo is in.

- 3 modes:
  - normal, ghost, noisy ghost
- Light detection switches between normal and ghost.

Diagram:

- Pleohoo
- ghost_level = ?
- main.p
  - Dino
  - Ghost
  - Noisy Ghost
- sensor.p
  - SENSOR_LIGHT
  - SENSOR_CHIRP
  - SENSOR_BACK
  - SENSOR_BACK_LEFT
  - SENSOR_BACK_RIGHT
  - SENSOR_FRONT
  - SENSOR_FRONT_LEFT
  - SENSOR_FRONT_RIGHT
- User Properties
  - (ghost_level, 0)
Two new behaviours for Pleo: Pleohoo

Pleo plays noisy ghost

The use of "user properties" in combination with sensors determine what mode Pleo is in.

- 3 modes:
  - normal, ghost, noisy ghost
- Light detection switches between normal and ghost.
- Lifting up Pleo in ghost mode switches him to noisy ghost.
- As soon as light is detected Pleo will switch back to normal.
Two new behaviours for Pleo: Pleohoo

Pleo plays noisy ghost

The use of "user properties" in combination with sensors determine what mode Pleo is in.

- 3 modes: normal, ghost, noisy ghost
- Light detection switches between normal and ghost.
- Lifting up Pleo in ghost mode switches him to noisy ghost.
- As soon as light is detected Pleo will switch back to normal.
- User properties are important for data exchange in Pleo’s interpreter script.
Two new behaviours for Pleo: Pleohoo

Pleo plays noisy ghost

The use of "user properties" in combination with sensors determine what mode Pleo is in.

- 3 modes:
  - normal, ghost, noisy ghost
- Light detection switches between normal and ghost.
- Lifting up Pleo in ghost mode switches him to noisy ghost.
- As soon as light is detected Pleo will switch back to normal.
- User properties are important for data exchange in Pleo’s interpreter script.
The main task of this thesis

Idea

To be able to switch Pleo’s behaviour during runtime a more complex behaviour had to be created in order to control Pleo via remote from an external server.
The main task of this thesis

Idea
To be able to switch Pleo’s behaviour during runtime a more complex behaviour had to be created in order to control Pleo via remote from an external server.

The protocol in theory - PRCP
- 2 layers of communication.
- Design of the packages.
The main task of this thesis

Idea
To be able to switch Pleo’s behaviour during runtime a more complex behaviour had to be created in order to control Pleo via remote from an external server.

The protocol in theory - PRCP
- 2 layers of communication.
- Design of the packages.
- Handling of different scenarios.
- Acknowledgement and three-way handshake.
The main task of this thesis

The interpreter script: Pleoslave

Different drives and behaviours manage the states Pleo can be in: Registration, IRSearch and Interpret.
The main task of this thesis

The interpreter script: Pleoslave

Different drives and behaviours manage the states Pleo can be in: Registration, IRSearch and Interpret.

- Orientate towards the IR source
- Retrieve data from the IR sensor
The main task of this thesis

The interpreter script: Pleoslave

Different drives and behaviours manage the states Pleo can be in: Registration, IRSearch and Interpret.

- Orientate towards the IR source
- Retrieve data from the IR sensor
- Registration: three-way handshake
- Check packages & interpret cmd

Limitation
Pleo's CPU power and memory is limited therefore it's important to reuse code as often as possible and avoid unnecessary memory allocations.
The main task of this thesis

The interpreter script: Pleoslave

Different drives and behaviours manage the states Pleo can be in:
Registration, IRSearch and Interpret.

- Orientate towards the IR source
- Retrieve data from the IR sensor
- Registration: three-way handshake
- Check packages & interpret cmd
- Send replies / ACK packages
The main task of this thesis

The interpreter script: Pleoslave

Different drives and behaviours manage the states Pleo can be in: Registration, IRSearch and Interpret.

- Orientate towards the IR source
- Retrieve data from the IR sensor
- Registration: three-way handshake
- Check packages & interpret cmd
- Send replies / ACK packages

Limitation

Pleo’s CPU power and memory is limited therefore it’s important to reuse code as often as possible and avoid unnecessary memory allocations.
The main task of this thesis

The Device: USB Infrared-Transceiver

In order to exchange data between Pleo and the server a USB IR-Transceiver had to be build that enables bidirectional communication.
The main task of this thesis

The Device: USB Infrared-Transceiver

In order to exchange data between Pleo and the server a USB IR-Transceiver had to be build that enables bidirectional communication.
The main task of this thesis

The Device: USB Infrared-Transceiver

In order to exchange data between Pleo and the server a USB IR-Transceiver had to be built that enables bidirectional communication.

- Especially built by the WWE Uni-Ulm.
- Handles the reception and sending of NEC packages.
The main task of this thesis

The Device: USB Infrared-Transceiver

In order to exchange data between Pleo and the server a USB IR-Transceiver had to be build that enables bidirectional communication.

- Especially built by the WWE Uni-Ulm.
- Handles the reception and sending of NEC packages.
- Returns and takes timings of IR-Impulses.
- Easy access via USB.
The main task of this thesis

The Device: USB Infrared-Transceiver

In order to exchange data between Pleo and the server a USB IR-Transceiver had to be build that enables bidirectional communication.

- Especially built by the WWE Uni-Ulm.
- Handles the reception and sending of NEC packages.
- Returns and takes timings of IR-Impulses.
- Easy access via USB.
- Didn’t work properly for a while.
The main task of this thesis

The Server side: Pleo Remote Control

This software is written in Java and constructed with a MVC pattern that enables the easy exchange of any component.
The main task of this thesis

The Server side: Pleo Remote Control

This software is written in Java and constructed with a MVC pattern that enables the easy exchange of any component.

- IR-Adapter & Decoder classes manage infrared communication.
- Classes holding data enable to monitor the state of all registered Pleos.
The main task of this thesis

The Server side: Pleo Remote Control

This software is written in Java and constructed with a MVC pattern that enables the easy exchange of any component.

- IR-Adapter & Decoder classes manage infrared communication.
- Classes holding data enable to monitor the state of all registered Pleos.
- An easy to use GUI with logging mechanism enhances usability.
- Runs under Windows as well as under Linux.
The main task of this thesis

The Server side: Pleo Remote Control

This software is written in Java and constructed with a MVC pattern that enables the easy exchange of any component.

- IR-Adapter & Decoder classes manage infrared communication.
- Classes holding data enable to monitor the state of all registered Pleos.
- An easy to use GUI with logging mechanism enhances usability.
- Runs under Windows as well as under Linux.
What does/doesn’t yet work?

1. Pleo finds and stays towards the IR source.

2. Pleo can register at server side and receives IDs.
What does/doesn’t yet work?

1. Pleo finds and stays towards the IR source.
2. Pleo can register at server side and receives IDs.
3. Pleo only interprets packages that are meant for him.
4. Send commands to Pleo via the PRC software.
What does/doesn’t yet work?

1. Pleo finds and stays towards the IR source.
2. Pleo can register at server side and receives IDs.
3. Pleo only interprets packages that are meant for him.
4. Send commands to Pleo via the PRC software.
5. Register two Pleos during one communication session.
6. The correct use of sequence numbers.
What does/doesn’t yet work?

1. Pleo finds and stays towards the IR source.
2. Pleo can register at server side and receives IDs.
3. Pleo only interprets packages that are meant for him.
4. Send commands to Pleo via the PRC software.
5. Register two Pleos during one communication session.
6. The correct use of sequence numbers.

1. Sending of acknowledge packages.
2. The use of timers on Pleo and server side.
What does/doesn’t yet work?

1. Pleo finds and stays towards the IR source.
2. Pleo can register at server side and receives IDs.
3. Pleo only interprets packages that are meant for him.
4. Send commands to Pleo via the PRC software.
5. Register two Pleos during one communication session.
6. The correct use of sequence numbers.

1. Sending of acknowledge packages.
2. The use of timers on Pleo and server side.
3. Sending commands in a setup with two Pleos.
4. Adjust parameters (waiting times, ...).
Setbacks and solutions

Theory vs. Practice

In theory, there is no difference between theory and practice.
But, in practice, there is.  

Jan L. A. Van de Snepscheut
Setbacks and solutions

Theory vs. Practice

In theory, there is no difference between theory and practice.
But, in practice, there is.  

Jan L. A. Van de Snepscheut

Inter-VM communication

- Pleo runs 4 different VMs:
  Sensor, Main, User, Script.
- Each VM has its own stack and heap.
Setbacks and solutions

Theory vs. Practice

In theory, there is no difference between theory and practice.
But, in practice, there is. Jan L. A. Van de Snepscheut

Inter-VM communication

- Pleo runs 4 different VMs:
  Sensor, Main, User, Script.
- Each VM has its own stack and heap.
Setbacks and solutions

Theory vs. Practice

In theory, there is no difference between theory and practice. But, in practice, there is.

Jan L. A. Van de Sneepscheut

Inter-VM communication

- Pleo runs 4 different VMs: Sensor, Main, User, Script.
- Each VM has its own stack and heap.

- Data needs to be shared and exchanged.

- Solution: "user properties"
Ideas for the future

▶ Outsource several tasks which Pleo can’t do.
▶ Scenarios with multiple Pleos and one server.
Ideas for the future

- Outsource several tasks which Pleo can’t do.
- Scenarios with multiple Pleos and one server.
- Use PleoRB and RFIDs.
Ideas for the future

- Outsource several tasks which Pleo can’t do.
- Scenarios with multiple Pleos and one server.
- Use PleoRB and RFID.
- Collecting data from Pleo’s sensors.
- Trigger user’s emotion through Pleo for research.
Summary

▶ Start off by choosing a sensor platform and analysing it.
▶ Find a way of data exchange & develop a protocol.
Summary

- Start off by choosing a sensor platform and analysing it.
- Find a way of data exchange & develop a protocol.
- Build two new distinguishable behaviours for Pleo.
- Write complex interpreter script handling PRCP.
Summary

- Start off by choosing a sensor platform and analysing it.
- Find a way of data exchange & develop a protocol.
- Build two new distinguishable behaviours for Pleo.
- Write complex interpreter script handling PRCP.
- Develop a flexible server software that can use PRCP.
- Pointed out setbacks and problems that slowed the process.
Summary

- Start off by choosing a sensor platform and analysing it.
- Find a way of data exchange & develop a protocol.
- Build two new distinguishable behaviours for Pleo.
- Write complex interpreter script handling PRCP.
- Develop a flexible server software that can use PRCP.
- Pointed out setbacks and problems that slowed the process.
- Talked about what is done and still needs to be done.
- Presented future ideas that can build upon this thesis’ work.
Any questions?

Feel free to ask.

Sources

- Manuel Meder: Entwicklung von anpassbaren Verhaltensmustern für eine autonome mobile Sensorplattform
- UGOBE LIFE FORMS: Pleo Programming Guide.
- Video Footage is supported by Thilo Hörnle and Peter Kurzok.
- http://blog.prinz.de/grand-prix/files/VIDEO_SYMBOL.png