Smart Indicator Environment

Christian Glahn
Overview

- Background, Purpose, and Objectives
- Context Model
- Smart Indicator Architecture
- Building and Processing Learner Profiles
- Context Adaptation
- Web Integration
Learning changes throughout life

... and the learner support has to change, too!
The Challenge

How to utilize interaction footprints for learner support in unstructured or emerging environments?
Our Approach

Highlight and unveil interaction footprints (about effort, interest, and concepts) for reflection support to the learner
The underlying Interaction-Model

Actor

Behaviour

Monitoring / Assessment

Response

Judgement / Reflection

Experience Knowledge

Interaction Footprints

System

Process Log

Reflection Support

Butler & Winne, 1995

Dey, 2000

(TEN Compete)
Example:
Context Adaptation for Informal Learning

Engage
Motivate
Reflect

actions
performance
interest
The Context Model
Core Elements

- Resources
- Context
- Boundary
Context Boundaries

• Constants
• User or Environmental Variables
  – User ID
  – Physical Location
  – URL
  – ...
• Aggregator References
• Conditions

• Results are always TRUE or FALSE
Adaptation Strategies
Adaptation Strategies

• Pre-defined or Static Strategies
  – Controlled
  – Driven by design principles

• Dynamic Strategies
  – On-the-fly combining of context definitions
  – Driven by
    • User preferences
    • Process dynamics
  – Combine static strategies
Think Adaptation Strategies the Web2.0 way

Parts and components that can be integrated and reused in unforeseen contexts

Limit design to context definitions
Why is it called “Smart Adaptation”?

- Context boundaries are rule based
- Adaptation depends only on Boolean operations
- No artificial intelligence is required
Smart Indicator Contexts

Sensor Reference

Context Boundary

Aggregator Reference

Indicator Reference
Explore and Engage

#blog-entries = 0

- click through
- delicious links
- page visits

every action counts
Compare and Motivate

- delicious links * 5
- blog entries * 10
- click through * 1

every action has its value for the user and the community
The Smart Indicator Architecture
Actual Implementation

- Selection Sensor (click-through detection)
- Tagging Sensor (del.icio.us JSON Interface)
- Contribution Sensor (RSS/ATOM Feed Reader)

Sensor Layer

Semantic Layer
- Activity Aggregator
- Interest Aggregator

Control Layer

Indicator Layer
- Interface Widget
- Interface Widget

Process Log

Aggregator Definition

Software Plug-ins

Service

Service

Service

AJAX Plug-in

HTML Injection

Data Flow

Style Sheets

Control
A Few Technical Details about the Services

- RESTful services
- implemented as LAMP (Linux, Apache, MySQL, and Perl)
- XML and JSON as output content types
- XML and JSON as input content types
Purpose of the Architecture

- Building Learner Profiles
- Processing Learner Profiles
- Context Adaptation

Sensor Layer
- Selection Sensor (click-through detection)
- Tagging Sensor (del.icio.us JSON Interface)
- Contribution Sensor (RSS/ATOM Feed Reader)

Semantic Layer
- Activity Aggregator
- Interest Aggregator

Control Layer

Indicator Layer
- Interface Widget
- Interface Widget

Process Log

Data Flow
- Control

Style Sheets
Building and Processing Learner Profiles

Sensor Layer

Semantic Layer
- Activity Aggregator
- Interest Aggregator

Process Log

TEN Competence
Building The European Network for Lifelong Competence Development
Sensor Layer

- Simple Sensors Registration
- Collects Interaction Footprints
- Extensible Sensor Events
- Sensor Event Clustering

The sensor layer is *not* a replacement of Log4J or similar debugging systems
Simple Sensor Registration

- Attach new sensors to the sensor layer
- Start to send events from the sensor
Collect Interaction Footprints

• Sources
  – RSS News Feeds
  – Del.icio.us Bookmarks
  – Online Web-interaction

• Sensors submit interaction events
• Interaction events are stored in the learner’s process log
Extensible Sensor Events

• Core data
  – Sensor ID
  – Referrer URL
  – Recoding Time
  – Referee URL (optional)
  – User ID (optional)

• Additional Attributes
  – Named and anonymous attributes
  – No limit of attributes
Sensor Event Clustering

• Collect events from multiple sensors at once
• Minimize network traffic
• Allow sensor caching, sensors proxies, or sensor cues
Semantic or *Aggregation* Layer

- Analyze the process log
  - Anonymous analysis
  - User centered analysis

- Open framework for sensor analysis
- Named aggregators
- Extension through aggregator scripts
Aggregator Scripts

![Diagram of Aggregator Scripts]

- **Result Data Set**
- **Operator**
- **Data Set Collector**
  - **Result Set Definition**
  - **Limits of the Data Set**
  - **Conditions**
- **Data Set Collector**
  - **Result Set Definition**
  - **Limits of the Data Set**
  - **Conditions**
- **Fixed Data Set**

*Note: TEN Competence logo and tagline*
Context Adaptation

Activity Aggregator
Interest Aggregator
Control Layer
Indicator Layer

Style Sheets

TEN Competence
Building The European Network for Lifelong Competence Development
Key functions

• Test context boundaries
• Provide a list of active contexts
  – For a user
  – For a location
  – For a URL
Test context boundaries

Request information from a given aggregator
  – Optional: with user information

Test the result set of the aggregator with another value

If the test succeeds the context is considered as active
Provide a list of active contexts

_Result set contains for each context_

• Context reference
• List of resource URL
  – Aggregator references
  – Indicator style-sheet references
Web Integration
Case: TeamSpace
Sensor and Indicator Integration

• Completely implemented as AJAX components
• Injection Code written in native Javascript
• Independent sensor and indicator components
Frontend Architecture

First level Services
(delicious/blog cache; shoutbox, search)

User Management and Registration

Sensor Service

Context Detection Service

Aggregator Service

TeamSpace

UserManager

Sensor Manager

Indicator Manager

EventManager
Purpose of the Architecture

Minimize interference with the business logic of a web-application

• Application independent code injection
  – Sensor Code
  – Indicator Code

• Modular frontend for web-applications
Sensor Manager

• Connects to the sensor service
• Requests registered sensors
• Registers supported sensors with the application
• Gathers user interface events
• Submits the collected events to the sensor service
Indicator Manager

- Connects to the context adaptation service
- Requests the active contexts
- Fetches visualization XSLT templates
- Prepares aggregator requests
- Handles HTML injection of the indicators
Management of Non-DOM-Events

Non-DOM-events = high level application logic

• Richer meaning of things that happen in the UI
  – Independent from the DOM structure of the UI
  – Connection points for events on code level

• Sub-systems can hook in on high-level functions
  – E.g. “followlink”, “userupdate”, or “sensorupdate”

• Events can be triggered by different sub-systems
EventManager

• Handling ambiguous DOM events
  – E.g. Follow link
• Assignment of application events to DOM events
• Entry point for sub-systems
• Event distribution across sub-systems
  – Following the principles of DOM events
Overview

- Background, Purpose, and Objectives
- Context Model
- Smart Indicator Architecture
- Building and Processing Learner Profiles
- Context Adaptation
- Web Integration
Thank you