A Whirlwind Overview of
The Global Public Inclusive Infrastructure

Colin Clark
Lead Software Architect,
Inclusive Design Research Centre
inclusive design institute

- Home
- About
- People
- News
- Research
- Infrastructure

- Design & Development
- Implementation & Information Practices
- Mobile & Pervasive Computing
- Business Case, Policies, Standards & Legislation
What is accessibility?
Rethinking Disability
Rethinking Disability

A mismatch between the user and the user interface
Accessibility is...

the ability of the system to accommodate the needs of the user
Make yourself at home...
Problems & Needs

The Internet is no longer optional, but many can’t use it.

Computers are everywhere, but accessibility isn’t.

The assistive technology market is stagnant.

The cost of assistive technology is unsustainable.

We are in a new era of cross platform diversity.
Personalization...

- **User interfaces** that can adapt to the needs, preferences, tastes of users—feel at home
- **Content** that can be delivered in a form that people can understand and use

... across all the platforms we use.
Envisioning the GPII
Cloud4all: Big Project, Little Steps

- Focus on a few use cases first: *Linux, Windows, and media on the web*
- Do everything in the open
- Get it working and iterate
Core GPII

- Automatic setup of assistive technology
- Adaptive web interfaces, services, and content
- Web, desktop, mobile
- Preferences stored where users want them: out on the web or close to home
Demo
Floe: Open Learning on the Web

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ut dui eu dui imperdiet fringilla. Fusce ac lectus quam.

Donec scelerisque diam a dui venenatis id pulvinar quam imperdiet. Ut eleifend, quam eget gravida rhoncus, nibh quam dapibus arcu, sed tristique justo ante vitae purus. Leo fermentum urna
Technical Goals

• **Lower the cost of building accessibly:** Developers can draw from a diverse range of easy to find adaptive building blocks

• Build **user personalization** into the fabric of the web, mobile, and desktop

• **Sustain** an infrastructure for sharing and interoperability across platforms
Technical Challenges

• How do we make this work across platforms? (Linux, Windows, Android)
• How does it work for web apps?
• How do we scale it big?
Solutions

• Use the web! REST and JSON payloads
• Don’t make new custom remoting APIs or use crusty old ones like CORBA
• HTML, CSS, JavaScript for user interfaces
• Deliver cross platform with Node.js + native bindings
Approach

- Language agnostic
- Ontology agnostic: go where users are
- Relocatable
- Scalable
- Adoptable
Preferences and Preferences Server

Accessible via:

- Preferences editor interface
- Web-based preferences server
- Other GPII components (Flow Manager)

a framework for preferences editing
Preferences Framework

Preferences App Configuration

Preferences App Configuration

Preferences App Configuration

Preferences Wizard

Game Preferences Editor

Immediate, inline Editor

Presentation

Persistence

Action

Resusable Preferences Objects

Tuesday, 19 June, 12
Preferences Server

Easy for developers to use

• RESTful API
• JSON-based payloads
• OAuth for delegating access

Scalable and forward-looking

• Document-based persistence
• High-concurrency server
Architecture Components

- Listen to user authentication requests
- USB, RFID, Barcode, etc.

- Information about the device
- Information about the environment
Architecture Components

- Matches the context to available solutions

- (a) - Registry of all the solutions for the device

- (b) - Transforms solutions into applicable configuration
Architecture Components

- Applies settings through Settings Handlers
- Launches AT applications through Lifecycle Handlers
Preferences & Settings

• User preferences are about people
• Application settings are custom
  • Different names, units, data types
  • Subset/superset
• Etc.
• Standard vocabulary has to be translated
Transformation Lenses

- Data-oriented
- Declarative and functional
- Bi-directional
- Useful collection of lenses out of the box
- Can be extended with new transformers
- AT integrators don’t have to write code
Preferences

{
    "display": {
        "screenEnhancement": {
            "fontSize": 24,
            "foregroundColor": "white",
            "backgroundColor": "black",
            "fontFace": {
                "fontName": ["Comic Sans"],
                "genericFontFace": "sans serif"
            },
            "magnification": 2.0,
            "tracking": "mouse",
            "invertImages": true,
            "showCrosshairs": true
        }
    }
}

Tuesday, 19 June, 12
“org.gnome.desktop.a11y.magnifier”: {
    “mag-factor”: 2.0,
    “mouse-tracking”: “centered”,
    "showCrosshairs": true
}

Tuesday, 19 June, 12
Settings (Windows)

```
"$HKEY_CURRENT_USER\Software\Microsoft\ScreenMagnifier": {
  "Magnification": 200, [REG_DWORD INT]
  "Invert": 1 [REG_DWORD BOOL],
  "FollowMouse": 1 [REG_DWORD BOOL]
}
```
Transform (Linux)

"capabilitiesTransformations": {
  "mag-factor": "display.screenEnhancement.magnification",
  "show-cross-hairs": "display.screenEnhancement.showCrosshairs",
  "mouse-tracking": {
    "expander": {
      "type": "fluid.model.transform.valueMapper",
      "inputPath": "display.screenEnhancement.tracking",
      "options": {
        "mouse": {
          "outputValue": "centered"
        }
      }
    }
  }
}
"capabilitiesTransformations": {
    "Magnification": {
        "expander": {
            "type": "gpii.transformer.scaleValue",
            "inputPath": "display.screenEnhancement.magnification",
            "outputPath": "value",
            "factor": 100
        }
    },

    "dataType": {
        "expander": {
            "type": "fluid.model.transform.literalValue",
            "value": "REG_DWORD"
        }
    }
},

...
Transformer Function

```javascript
var factor = expandSpec.factor === undefined ? 1.0 : expandSpec.factor;
return value * factor;
```
Breaking it Down
EL Paths

```json
{
  pets: [
    {
      type: "cat",
      name: "sirius"
    }
  ]
}

"pets.0.name"
```
Value Transform

{"<output path>": {
    "expander": {
        "type": "fluid.model.transform.value",
        "inputPath": "<input path>"
    }
}}
Value Transform Short

"<output path>" : "<input path>"
"<output path>": {
  "expander": {
    "type": "fluid.model.transform.literalValue",
    "inputPath": "This will literally be the output."
  }
}

Tuesday, 19 June, 12
Value Mapping

"mouse-tracking": {
    "expander": {
        "type": "fluid.model.transform.valueMapper",
        "inputPath": "display.screenEnhancement.tracking",
        "options": {
            "mouse": {
                "outputValue": "centered"
            },
            "caret": {
                "outputValue": "none"
            }
        }
    }
}
Custom Transforms

"Magnification": {
    "expander": {
        "type": "gpii.transformer.scaleValue",
        "inputPath": "display.screenEnhancement.magnification",
        "outputPath": "value",
        "factor": 100
    }
}
Reversible: Settings-Oriented

```json
{
    "mag-factor": "display.screenEnhancement.magnification",
    "show-cross-hairs": "display.screenEnhancement.showCrosshairs",
    "mouse-tracking": {
        "expander": {
            "type": "fluid.model.transform.valueMapper",
            "inputPath": "display.screenEnhancement.tracking",
            "options": {
                "mouse": {
                    "outputValue": "centered"
                }
            }
        }
    }
}
```
Reversible: Preferences-Oriented

```json
{
  "display.screenEnhancement.magnification": "mag-factor",
  "display.screenEnhancement.showCrosshairs": "show-cross-hairs",
  "display.screenEnhancement.tracking": {
    "expander": {
      "type": "fluid.model.transform.valueMapper",
      "inputPath": "mouse-tracking",
      "options": {
        "true": {
          "outputValue": "mouse"
        }
      }
    }
  }
}
```
Basic Matching

• In the solutions registry, transform rules are “settings-oriented”

• Flip them around and they’re “user-oriented”

• Match on fine-grained setting/preference correspondence (no hierarchy, inheritance, formal ontology etc. required)
Source Code

https://github.com/GPIII/

Example code and configuration:

• Settings Handler: http://bit.ly/M6mDfD
• Solutions Registry: http://bit.ly/KrLo6Z