Proposing a Meta-Language for Specifying Presentation Complexity in order to Support System Situation Awareness
Situation Awareness

- “knowing what is going on around you”
- Automotive Domain: Helps us reduce accidents
- Subgoal: Reducing distraction
- System Situation Awareness
- Endsley Model
Endsley Model (original)

**Task/System Factors**
- State of the Environment
- Decision
- Performance of Action

**Individual Factors**
- Goals, Objectives, Preconceptions (Expectations)
- Information Processing Mechanisms
- Long-term Memory Stores
- Automaticity
- Abilities, Experience, Training

**SITUATION AWARENESS**
- Level 1: Perception
- Level 2: Comprehension
- Level 3: Projection

**System Capability**
- Interface Design
- Stress & Workload
- Complexity
- Automation

**Feedback**
Endsley Model (adjusted)

Task/System Factors

- State of the User / Driver
- System Capability
  - Interface Design
  - Complexity
  - Automation

System Situation Awareness

- Level 1: Assessment of User and context
- Level 2: Updating Information Sources
- Level 3: Impact Estimation

Intelligent Mediation

Individual Factors

- Goals, Objectives
  - Preconceptions (Expectations)
- Information Processing Mechanisms
  - Long-term Memory Stores
  - Automaticity
- Abilities
  - Experience
  - Training
Two-fold Research Question

Driver Aspects

Defining System Situation Awareness

Driver related concepts

Assessing Cognitive Load

Upgrading User Profile with CL

Achieving System Situation Awareness

Presentation Aspects

Presentation Task Annotation

Estimating Complexity

Presentation Meta Language

Updating User Profile with CL

Defining System Situation Awareness

Preparation

Level 1

Level 2

Driver related concepts

Assessing Cognitive Load

Upgrading User Profile with CL

Achieving System Situation Awareness
Estimating Presentation Complexity

- Three Options:
  - Complexity specified by designer
    - “Ideal” case $\rightarrow$ nothing to do
  - Unstructured representation
    - Heuristic approaches $\rightarrow$ low confidence
  - Structured representation (e.g. HTML5)
    - ACE (Annotated Complexity Estimation)

- Third case:
  - How to annotate complexity automatically?
  - ACE based on visual tree and complexity table
Example Screen Layout (sim^{TD})
GUI Model and Visual Tree

![Diagram of GUI Model and Visual Tree](image-url)
Complexity Computation

```
0.8  iconpanel
    empty icon
    ... empty icon
    0.1

0.1  named panel 3.4
    label
    with icon
    1.0

1.0  named panel 2.4
    label
    with icon
    1.0

1.0  unnamed panel 1.7
    label
    with icon
    1.0

1.0  button
```

### Table: Component Complexity

<table>
<thead>
<tr>
<th>component</th>
<th>basic complexity</th>
<th>feature</th>
<th>added</th>
</tr>
</thead>
<tbody>
<tr>
<td>label</td>
<td>0.1</td>
<td>text=true, icon=true</td>
<td>+0.5</td>
</tr>
<tr>
<td>icon</td>
<td>0.1</td>
<td>type=empty, type=icon, type=icon</td>
<td>+0.0,  +0.5</td>
</tr>
<tr>
<td>panel</td>
<td>0 + ∑ child nodes</td>
<td>decoration=framed, decoration=none, metainfo=named, metainfo=none</td>
<td>+0.2,  +0.5,  +0.2,  +0.5</td>
</tr>
</tbody>
</table>
Presentation Meta Language

- Developer can provide multiple presentation alternatives
- System can choose based on complexity and driver workload
- Goal: No new presentation language

→ Wrapper or **Meta Language**

```xml
<ptcl>
  <meta>
    <overallPriority value=70 metric="percent" />
  </meta>
  <displayStrategies>
    <strategy>
      <preference=1 />
      <demand=0.8 />
      <representation language="XY">
        [first variant of presentation task in language XY]
      </representation>
    </strategy>
    <strategy>
      <preference=2 />
      <demand=0.3 />
      <representation language="XY">
        [second variant of presentation task in language XY]
      </representation>
    </strategy>
  </displayStrategies>
</ptcl>
```
Implementation into a Dialogue Platform

Situation-Adaptive Multimodal Dialogue Platform

SCXML-based
Dialogue Offline Evaluation

- Modeling dialog cost (metrics)
  - Cognitive load
  - Time
  - Usability
  - Money
  - Total cost

- Anticipating the cost of a dialog already at design time (without expensive user study)
  - Expected cost on given path
  - Most costly transitions
  - Shortest / longest path
  - Average path
  - Best modality / modality comparison
Estimating (Input) Interaction Workload

Estimating interaction cost

Analyzing the dialog model and task complexity

Breaking up complex tasks into atomic tasks

determined in separate studies

<table>
<thead>
<tr>
<th>Rearrangement</th>
<th>Text entry</th>
<th>Number entry</th>
<th>Pan / Zoom</th>
<th>List selection</th>
<th>Scrolling</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>text</td>
<td></td>
<td></td>
<td>list of text</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Widgets</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Touchscreen, Speech, Eyegaze, Micro-gesture
Two-fold Research Question

- Driver Aspects
- Presentation Aspects

Defining System Situation Awareness

- Driver related concepts
- Asessing Cognitive Load
- Updating User Profile with CL

Level 1

- Preparation
- Task Annotation
- Estimating Complexity

Level 2

- Presentation Meta Language

Achieving System Situation Awareness
Driver-related Cognitive Load Aspects

Main Questions:

• How to **model** cognitive load?

• How to **quantify** cognitive load?
Cognitive Load Model

System Interaction

“Dry” Demand
situation-independent e.g. from lookup table

Situation-adjustment (e.g. time)

Cognitive Demand

Driver

Cognitive Capacity

Effects

distraction, stress,…

Total Cognitive Load

Cognitive Resources

Auditory Processing Resources

Visual Processing Resources

Cognitive Cost

Tasks / Stimuli

Cognitive Demand

Cognitive Load

Stress / Distraction

Metrics

Time, accuracy, driving performance, pupils, biosensors,…

Dialog System

e.g. situation-independent

Situation-

“Dry” Demand

Metrics

"Dry" Demand

Auditory Processing Resources

Visual Processing Resources

Tasks / Stimuli

Cognitive Demand

Cognitive Load

Stress / Distraction

M. Feld
Cognitive User Model

- User
  - ProcessingResource (1..n)
    - Dimension
    - CognitiveCapacity
    - CognitiveCost (1..n)
      - Amount

  - Processing Stage: Perception / Cognition
  - Modality: Visual / Auditive / ...
  - Visual Channel: Focal / Ambient
  - Processing Code: Spatial / Symbolic

- Context
  - Stimuli (1..n) (permanent)
    - GetCurrentCognitiveDemand() : CognitiveDemand
  - Interaction (only temporarily present)
RELATED EFFORTS
Automotive Ontology

On the one hand...

Knowledge in the Modern Car

- Sensors & Controls
  - Inside
  - Outside
- Geographical Knowledge
- Traffic Management
- OEM Uplink
- Car2car
- Roadside Units (car2x)
- Internet Services
- Passenger Profiles
- Driving Habits
  - Roads, times, driving styles...
- Personal Devices
- ...

"Information Hub"
...and on the other hand

Feature-rich In-car Applications

- Driver Assistance
- Navigation
- Parking Assistance
- Comfort Controls
- eMail, SMS
- Twitter, Instant Messaging
- PIM
- News
- Information Search
- Entertainment, Music
- Navitainment
- Local Information
- ...

M. Feld
High-Level Structure

View from the users’s perspective

AutomotiveWorld

User

Context

Basic Dimensions
Interactions
Preferences
Presentation

Vehicle
Devices
External Physical
Trip
User Model

User

BasicDimensions

MentalState
- timePressure
- cognitiveLoad
- irritation
- trauma

Personality
- extraversion
- agreeableness
- conscientiousness
- neuroticism
- openness

Abilities
- canSee
- canHear
- canSwim
- sight
- hearing

Characteristics
- talkative
- assertive
- dominant
- quiet
- thorough
- helpful

PhysiologicalState
- heartbeat
- bloodPressure
- arousal
- fatigue
- alcoholLevel

EmotionalState
- happiness
- anxiety
- anger
- disgust
- sadness
Meta Information

Time

Location

Confidence

Privacy
Driving Simulation

- We created a new 3D Driving Simulator in order to measure the driver’s distraction in a controlled lab environment.
- The simulator is connected via sockets with the HMI that displays important information about the upcoming road segment.
- The screens show examples from simTD.
  - **Road Works Information**: a progress bar is shown and the distance counts down until the construction site is reached.

When the construction site is reached, the current position is shown and the time to reach the end elapses.

Dynamic objects supported!
Driving Performance Measures

- The Simulator can record the driven path as a list of way points.
- In the Drive Analyzer, this path can be compared to a predefined “ideal line” by computing the average deviation.
- The smaller the area between both lines, the higher the driving quality (c.f. evaluation of Lane Change Test).
- The new 3D Driving Simulator with the shown features is now able to simulate the Lane Change Test from the beginning.
- Arbitrary map models can be loaded (as long as they can be processed with Blender).
- The physics simulation is based on a realistic car.
- Triggers to hide/show lane signs can be placed.
- Evaluation after drive with common “deviation computation” approach.
- This approach can be modified and extended to our future needs.

Drive Analyzer in top view and chase camera view. The pink line denotes the ideal path and the yellow line the driven path.

distance more than 40 meters: hidden signs
distance less than 40 meters: visible signs
Traffic Light Control

- Fully controllable traffic lights
- Traffic light programs
  - Triggered traffic light control: only if a car approaches to an intersection the corresponding traffic light will be requested to turn green
  - Internal traffic light control: a given list of traffic light phases will be processed
  - External traffic light control: the simulator waits for external traffic light status inputs (either manually or by a 2D traffic simulator like SUMO)

instruction sent by SUMO

```
<TrafficLightControl>
  <tlsstate timeR="178.00" id="0" programID="0" phase="6" state="grrrgrrr"/>
</TrafficLightControl>
```
External Visualization

- For the new 3D Driving Simulator a special model of Saarbrücken was created as a part of the “Stadtmitte am Fluss” model (by DFKI’s agents and simulated reality group)
- Original map data has been provided by the land registry (Landesamt für Kataster-, Vermessungs- und Kartenwesen)
- Extended by street data extracted from the Open Street Map project
- Simulator computes geo-position to show in Google Maps

traffic light states, camera position and orientation
More Information on OpenDS

www.gethomesafe-fp7.eu

THANK YOU!