ETAG – extended task-action grammar

formalism (Tauber, 1990)
task-action grammar (TAG) with an ontology
psychologically valid wrt user mental model
model for design specification
... and possibly more

Task-Action Grammar
(TAG; Green and Payne, 1986)

Task [Direction, Unit] -> symbol [Direction] + letter [Unit]
symbol [Direction = forward] -> “>“
symbol [Direction = backward] -> “meta“
letter [Unit = character] -> “C“
letter [Unit = word] -> “W“

ETAG Sources

Design specification (Moran 1981)
- Ontological Level
- Conceptual Level
- Semantic Level
- Syntax Level
- Lexical Level
- Physical Level

Psychological validity
- conceptual graphs (Sowa, Klix ...)
- competence model
  not a performance model
  user mental model of a perfectly knowing user

ETAG-based Design

• task analysis
• functional design
• conceptual design
• perceptual design
• implementation

• one notation
• most iterations
within design steps &
few between

ETAG the Model

[CONCEPT] ::= [OBJECT] | [VALUE] | [PLACE] | [STATE] | [EVENT] ... 

[EVENT] ::= [event.KILL.ON (OBJECT), [PLACE]]
[EVENT] ::= [event.MOVE.TO (OBJECT), [PLACE]] ...

type [OBJECT > MESSAGE],
themes: [HEADER], [BODY],
relations: [place.ON.POS(1) (MESSAGE)] for [HEADER],
[place.POSS(1) (MESSAGE)] for [BODY],
attributes: [RECEIVING_DATE], [SENDING_DATE],
[OBJECT > MESSAGE], [EVENT > COPY_MESSAGES],
[MESSAGE_FILE: *z]

"copy messages from the current message file into a message file".
ETAG as a Formal Model

advantages of a formal design specification:
- automatic generation of programming code
- generation of help text
- calculation of usability metrics
  using ontology and state-history information!
- a single notation throughout design
  only one specification language to master
- formality failed for the perceptual interface
  we cannot generate suitable screen/physical designs

Current methods and tools

Media Technology products: websites, interactive webs, mobile apps, CMS-es: frontend website, backend CMS & a database, code-glue

Pragmatic methodology around UCD & prototyping
- design concept, design view
- persona's (mood boards)
- scenario's, storyboards
- task lists, task descriptions (task analysis)
- use cases, activity diagrams, ER diagrams
- interface sketches, wireframes
- (paper) prototypes, demonstrators, actual designs

Questionnaire
- first inventory to set up followup research
- students who work as interns for a BA thesis
- name (or anonymous)
- which company
- type of product (cms-es, websites, mobile apps, ads ...)
- main methods (Scrum, SDM/structured, waterfall ...)

about 21 students, 11 respondees: almost all scrum, waterfall (xp, ad hoc, dsdm)
ergo: lightweight, prototyping/incremental, with project management

Formal models and Media Design

Rotterdam University of Applied Sciences
BA-level Media Technology
(engineers not scientists!)
Research group Human-Centered ICT - - which methods should we teach with respect to
- optimal support for current technology (webs, cms-es ...)
- future technology (ubicomp, sentient systems ...)
- the companies who hire our students
Investigate: current teachings, companies, software & possible future developments

Companies – Pilot Inventory

- ads, websites, web services, web & mobile apps
- most young, small, hip, innovative
- development methods
  - soft requirements
  - flexible methods for e.g. social media, new products, ubicomp
  - experiences with methods practice
- which methods do 'our' companies use?
  - SE: agile, structured, ucd, XP, RAD, RUP ...
  - non-SE: prince II, CMM, ...
  - type of company or department / products / size ...

Software products

frontend – backend systems
- frontend = presentation / user interface
- backend = database / functionality

flexible architecture (Seeheim model; 1984) ->
  'natural' support for UCD, incremental, empirical, adaptive, prototyping-based development
- even: development continues into production use

ergo: formal specification are not needed, unwanted
The Sentient Future

from SE to html, java, cms, smartphone, html5
the future is sentient, context-sensitive, adaptive
  ➔ micro-adaptive not model-based
  ➔ continuously changing user wants & needs

design requirements:
  ➔ close(s) contact with actual users
     co-design (& co-creation)
  ➔ facilities for quick conceptual prototyping
     sensor lab -> fablab
  ➔ facilities for continuous fine-tuning in the field
     living lab (cf. sense-os)

Software Design and Media Design
formal and non-formal tools

Geert de Haan
g.de.haan@hro.nl