Conceptual Modeling of Internet Sites
Part 2.3.1: Adaption of a website to portfolio, profile, and context

Conceptual Modelling of Internet Sites
Wintersemester 2014, Gastprofessor@FH Hagenberg, Österreich

16.10.2014

Prof. Dr. Bernhard Thalheim
Technologie der Informationssysteme
Institut für Informatik, Christian-Albrechts-Universität zu Kiel, BRD
Kolmogorow-Professor e.h. der Lomonossow-Universität Moskau
Fine Tuning?

- general approach: consider partial description of the entire site
- this includes mini-stories and associated media types as well as the database and database process support
- consider all descriptions to be incomplete (missing details, preliminary status, etc.)
- variation: discuss alternatives to the story leading to modified / extended mini-story
- add additional mini-stories and consider possible presentations of scenes
- refinement-validation cycle: refine a description & validate whether it meets the intentions
- consider effects of changes on media types
- goal: design a consistent set of media types, containers, user types, roles as a basis for site production
Refinement

- refinement in general means to replace an overall description of a site by a new one such that the old one can be derived from the new one

- a story was defined as a sequence of actions:
  - variation of the story means to add alternatives
  - detailing the story means to replace scenes by substories

- a mini-story was defined as a graph of scenes (actions):
  - associated with user-types, communicated information, consumed information
  - supported by media types (and indirectly by the underlying database)

- emphasize sequencing, communication, support or container in the presentation of mini-stories
Refinement

a) emphasize sequencing

b) emphasize communication

<table>
<thead>
<tr>
<th>interest in red grape</th>
<th>catalogue</th>
<th>grapes</th>
<th>grapes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>catalogue</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>addit.</td>
<td>grape</td>
<td></td>
</tr>
<tr>
<td></td>
<td>information</td>
<td>list</td>
<td></td>
</tr>
<tr>
<td></td>
<td>offers</td>
<td>prices</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>order</td>
<td></td>
</tr>
</tbody>
</table>

c) emphasize container / escort information

d) emphasize support

<table>
<thead>
<tr>
<th>interest in red grape</th>
<th>catalogue</th>
<th>addit. information</th>
<th>offers</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>catalogue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>addit.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>information</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>offers</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>grape</th>
<th>grape producer</th>
<th>grape yard</th>
<th>grapeyard region</th>
<th>offer</th>
<th>offer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>grape producer</td>
<td>grape yard</td>
<td></td>
<td>bottle</td>
<td>client</td>
</tr>
</tbody>
</table>
Mini-Story Refinement

- top-down primitives:
  - decompose scenes introducing variants or sequences
  - specialise scenes with / without replacement
  - add new scenes
  - adapt user types, communicated / consumed information, support

- bottom-up primitives:
  - aggregate substory into a single scene
  - generalize scenes with / without replacement
  - introduce user-types, roles, supporting media types, etc.

- summarization:
  - mark possible containers
  - mark escort information
  - identify candidates for media objects
  - reconsider the story to detect enabled / disabled constraints
Role and User Refinement

• specialize / generalize roles:
  • enable mini-story splits (one role per mini-story)
• obtain finer granularity for the user grid (set of user profiles) $gr(\Delta)$:
  • enlarge the scale $sc(\delta)$
  • additional user-types / hierarchy of user-types
• each scene is to be associated with a unique user-type
• reconsider stories constrained to user-types
Media Type Refinement(1)

- Media type refinement originates from tuned mini-stories with marked containers: enable loadability
- Media type refinement originates from refined user types
- Content primitives:
  - Split content data type for media types and introduce links
  - Combine / aggregate media types and remove links
  - Specialize (extend) content data type with / without escort information
- Functionality primitives:
  - Add / disable / specialize operations
  - Declare operations to be ‘add-on’ or use proximity values to indicate the grade of desirability
Media Type Refinement (2)

- scaling primitives:
  - add / remove dimensions: impact on hierarchical versions
  - add / refine cohesion / adhesion: finer adhesion preorder, added proximity values
  - switch to ordering / measuring

- database support primitives:
  - build database views, view integration, view cooperation
  - develop the underlying database schema

- reconsider mini-stories: what is supported? which communication is covered?
Container Refinement

- adapt container parameters:
  - extend / reduce capacity
  - specialize / generalize the content data type
  - consider loadability of container
    - after refining a container
    - after refining a media type
  - compare with mini-stories

- formation:
  - presentation mini-story on the basis of existing media types
  - check loadability, unloadability for versions (adhesion / cohesion, hierarchical versions)
  - check loadability with respect to figures

- ...
Container Refinement

- ... 
- wrapping:
  - consider the specific user needs originating from user-types 
  - wrapping rules are used to pack containers 
- styling:
  - develop the layout of the container 
  - refinement leads to more detailed presentation
Modeling Specific Operating Steps

- Decide what to do as next
- Specify next steps, aims and occasions
- Planning of operating actions
- Acting according to the plan
- Observation and recognition of systems reactions
- Interpretation of the observed systems state
- Comparison of interpreted and expected systems state
Design Decisions

- refinement is goal-oriented:
  - structure the site in a way that intentions are met in an optimal way
  - structure content, functionality, usage, presentation
- meta-level: organize the site, not just individual pages resulting from instantiations of media types
- use *structuring schemata*: common characteristics of pages
  - *information structure*: which pages (entities) with which relationships
  - *navigation structure*:
    - Hänsel & Gretel Syndrom: loss of orientation
    - how to find required information (browsing / overview)
  - *search structure*: give me . . .
Structuring Schemata

- purpose: structure and order information:
  - arbitrary with ambiguities, ellipses, definitions
  - heterogeneous (e.g., granularity, formats, content)
  - perspective-driven (e.g., client support)
  - intension-driven (e.g., vendor policy, marketing)

- choice of order:
  - order criteria: subject, tasks, usage, metaphors
  - fairly random (e.g., vendor policy) or clearly defined, easily understandable order (e.g., telephone book)
  - combine different orders

- ...
Structuring Schemata

- ...
- classification:
  - mono- or polydimensional, mono- or polyhierarchies
  - analytical versus synthetical classification
- use of a thesaurus
Information Structure

- goal: structure the content, though often determining the navigation as well

- functionality is considered to be attached (see definition of media types)

- hierarchical structures:
  - easy to achieve, but difficult to use
  - possible use in catalogues, ontologies, etc.
  - includes directed acyclic graphs

- ...
Information Structure

- ... 

- hypermedia structures:
  - highly flexible, but risk to lose orientation
  - the conceptual model relies on infinite (rational) structures and is difficult to understand

- underlying database structures: determinant by different design desiderata (non-redundancy, consistency, fast accessibility, etc.)
Navigation Structure

- goal: help user to keep orientation within a site
- do not give up flexibility goal
- navigation systems:
  - hierarchical navigation via simple backtracking
  - global navigation to enable general vertical and horizontal navigation
  - local navigation extending the global one in a closed subarea
  - ad-hoc navigation via meaningful anchors and hyperlinks
- ...
Navigation Structure

- ... 

- navigation aids:
  - explicit map (floor plan, structuring schema) of the site as long as this is two-dimensional
  - separation of pages into primary information and navigation information
  - use of a content description
  - use of an index / catalogue

- markups in a navigation system: headers, meaningful anchors, meaningful icons, keywords
Bulk Navigation

- Representation of navigation facilities by hierarchical (forest) structures with prerequisites and accept-on steps

- Generation of walk-through plans depending on profile of the user, on the environment currently used, connectivity on hand e.g., breath-first walk, depth-first walk, adhesion-directed walk, metaphor-based walks

- Comprehension of paths to units depending on the size of the data objects derivation of folders, subfolders etc. depending on the step currently under consideration

- Generation of linked units with (de-foldable) nested extensions depending on prerequisites and conditional dependencies defined by the completion status

- Composition of the site suite definition of criteria for completion status of a dialogue scene (e.g., exercise completed, exercise partially incorrect)

- Derivation of the (context-sensitive) navigation tree with time-stamps, completion-marks
Extending Navigation by Exploration Techniques

- Development of the exploration techniques for complex navigation spaces
  - Fish-eye view techniques (center in scope, rest compressed)
    - 3D-fish-eye techniques
    - adorned fish-eye views
    - fish-eye views with transformation of coordinates (radial (locally or globally), orthogonal (locally or globally), 3-dimensional (implicitly or explicitly))
    - filtered fish-eye views (hierarchically or by graph structures)
  - Zoom navigation techniques
  - Non-linear visualization of navigation based on focus points or by multi-point hyperbolic representation
  - Adopting size by weight functions (measures of relevancy or importance, size of data)
- Development of visualization alternatives
Tracking and Tracing of History

- Extending bookmark facilities to search profiles with
  - automatic lazy adaptation to the actual search behavior
  - similar to Knuth-search trees with weight adaptation
  - overlay structures for highlights
  - extended self-reminding facilities (EPG)

- Generation of sophisticated logs for user utilization history with shared history on the basis of history forests
  - reuse of existing related history tracks of other users

- Adaptation of site suite to history of successfully visited dialogue scene with completed tasks

- Application of learning algorithms for extension of profiles

- Undo/redo facilities for later recovery

- Import of generalized search profiles of other groups (people looking for this information asked for ...)

Integrating Various Kinds of Navigation

- Internal navigation through the internal database
  - Functional navigation on the basis of search/retrieval requests
    - search according to properties
    - solution-directed search
  - Vertical navigation for browsing through results of the retrieval
- Horizontal navigation on the basis of search/retrieval history
  - Navigation through the workspace of the client
  - Depending on the profile of the client
  - Forward/backward within the space
- ...
Integrating Various Kinds of Navigation

- Vertical navigation within the provided page
  - Navigation through properties
  - Navigation on the basis of browsing facilities of the page, e.g., mouse, roll-up/down
- External navigation to external sites and pages
  - On the basis of portal technology, e.g., via provided links
  - On the basis of query forms, concept maps and ontology
Retrieval and Search

- goal: support users to find specific information:
  - ideal: system acts like a skilled, experienced librarian
  - usage factors:
    - ability to formulate search query
    - precision of expectation

- ...

Retrieval and Search

- ... 

- search-engine for the site?
  - only necessary, if there is a frequently changing content as in database-backed sites
  - can otherwise be replaced by sophisticated browsing / navigation

- user support:
  - explanation: connectives, string versus keyword, scope of search, etc.
  - feedback: recall, precision as in Information Retrieval
  - integration of search, browsing and navigation aids
Media Object Query Algebra

Operations are defined on the basis of structural recursion \( op = \text{src}[e, g, \sqcup] \)
with a value \( e \) of type \( t' \), a function \( g : t \to t' \) and a function \( \sqcup : t' \times t' \to t' \):

\[
\begin{align*}
\text{src}[e, g, \sqcup](\emptyset) & = e \\
\text{src}[e, g, \sqcup](\{x\}) & = g(x) \quad \text{for } x \text{ of type } t \\
\text{src}[e, g, \sqcup](X \cup Y) & = \text{src}[e, g, \sqcup](X) \cup \text{src}[e, g, \sqcup](Y) \quad \text{for } X, Y \text{ of type } \{t\},
\end{align*}
\]
e.g., \( \text{filter}(\phi) = \text{src}[\emptyset, \text{if}_\text{then}_\text{else} \times (\phi \times \text{single} \times (\text{empty} \times \text{triv})), \sqcup] \)

\[
\text{sum} = \text{src}[0, \text{id}, +]
\]

Operations for tuple types are defined by structural recursion on basic operations
projection \( \pi_i : t_1 \times \ldots \times t_n \to t_i \), (Cartesian) product \( \times : t \to t_1 \times \ldots \times t_n \),
reordering \( \rho \), renaming \( \kappa \)

Operations for set types are defined by structural recursion on basic operations union \( \cup \),
difference \( \setminus \), constant, singleton element,
e.g., join operation

Operations on function types are defined by structural recursion on basic operations composition \( \star : (t_2 \to t_3) \times (t_1 \to t_2) \to (t_1 \to t_3) \), evaluation \( \text{ev} : (t_1 \to t_2) \times t_1 \to t_2 \)
and abstraction \( \text{abstr} : (t_1 \times t_2 \to t_3) \to (t_1 \to (t_2 \to t_3)) \)

Conversion operations from tuple to set types, from function types to collections, etc.

Operations for URL extension using labels \( l \) extending the type system to

\[
t = b \mid l \mid t_1 \times \ldots \times t_n \mid \{t\} \mid [t] \mid l : t
\]
that is restricted to rational trees (number of different subtrees is finite)
Classification

Classification \( A \): \((O_A, C_A, \in_A)\) for a set of objects, a set of classes (including the empty class and the full class of all objects) and an association of objects to classes

Infomorphism of classifications \( A_1, A_2 \) given by mappings \((f, g)\) with \( f : O_{A_2} \to O_{A_1} \) and \( g : C_{A_1} \to C_{A_2} \) such that \( f(o_2) \in_{A_1} c_1 \) iff \( o_2 \in_{A_2} g(c_1) \)

meaning of \( A_1 \geq A_2 \): \( A_2 \) may have less objects and classes, is of less granularity than \( A_1 \)

if an object belongs to several classes in \( A_2 \) then the corresponding object in \( A_1 \) also belongs to several classes that are mapped onto those in \( A_2 \)

Classification (lattice) theory \( T_A \): \((C_A, \sqsubseteq_A)\) for a set of classes and a set of generalized disjunctive inclusion constraints \( \Gamma \sqsubseteq \Delta \)

\[ c_1 \cap \ldots \cap c_k \subseteq c_q \cup \ldots \cup c_m \]

(expressible are: subtyping, partition, disjointness, covering, incoherence)

regular theory with identity, weakening, cuts,

partition (if \( \Gamma' \sqsubseteq \Delta' \) for a partition \( P \) of \( C_A \) with \((\Gamma', \Delta') \leq (\Gamma, \Delta) \) then \( \Gamma \sqsubseteq \Delta \))
Classification

... Local classification model $M_A$: $(O_A, C_A, \in_A, \subseteq_A, N_A)$ with the set of normal objects $N_A \subseteq C_A$ satisfying the constraints

sound logic if $N_A = O_A$

Associations between models, classifications and lattice theory: For the classification mapping $cla : \text{Model} \rightarrow \text{Classification}$, $Th : \text{Model} \rightarrow \text{Theory}$

For $M \in Mod(A, T_A)$: $cla(M) \geq A$, $Th(M) \supseteq T_A$
Ontology

Ontology is a classification with a set of constraints representing subtyping, partition, disjointness, covering and incoherence, more specifically

- set of terms (objects) of interest in a particular domain \( O = \{ o_1, \ldots, o_n \} \) and relationships \( \mathcal{R} = \{ R_1, \ldots, R_m \} \) among them (ontological commitment) relating concepts with kinds, valuation (value, modality, existency) and actors (worlds)
- \( o_i = (k, id_i, v_i) \) for \( k \in \text{Kind} \), \( v_i \) value form \( \text{DOM}(O) \), \( id_i \in \text{ID} \)
- \( R_i = \{ r_j = (tr_j, o_{j,1}, \ldots, o_{j,k}, o_j) \mid tr_j \in \text{TR} \}, o_{j,l} \) parameters of \( r_j \)
- \( \text{Kind} = \{ \text{predicator thing, action, actor, rule} \} \) (predicator/structural view; actor+action/dynamic view; rules/deontic view)
- \( \text{TR} = \{ \text{execute, actand, use, extend, ...} \} \) (dynamic view, e.g., actand )

Shared ontology of two communities \( G_1 \) and \( G_2 \) with \( A_1 \) and \( A_2 \) defined by:

Common generic extensible ontology \( A \) that can be mapped by infomorphisms \( (f_1, g_1) \) to \( A_1 \) and by \( f_2, g_2 \) to \( A_2 \)
greatest consistent classification that is finer (or equal) than \( A_1 \) and \( A_2 \)
Core classification \( A^* \) of the communities is defined by the fusion of the classification lattice theories of \( A_1 \) and \( A_2 \) modulo synonyms of \( A_1 \) and \( A_2 \), respectively
with a local classification theory (coincide on common classification, local on non-shared classifications)
Concepts

Query form - given by an

parametric algebraic expression and a

linguistic utterance expressing the meaning.

Concept map is defined by a set of objects and a set of relations \( R \) among concepts
(e.g. \( \text{PartOf} \), \( \text{Specializes} \), \( \text{IsA} \), \( \text{IsRequiredBy} \), \( \text{IsVersionOf} \)).

A concept is defined by

name identifying the concept and

query forms describing the retrieval environment of the concept.

Ontology objects are associated with a set of concepts.

Schema association of a concept is one instantiated query form of the concept corresponding to the database schema.

\[
\text{Author} = \text{BookAuthor} \oplus \text{ConferenceAuthor} \oplus \text{PreprintAuthor}
\]

\text{ConferenceAuthor}

\( \text{IsA} \) (Author of a publication in a conference)

\( \text{IsA} \) (Author IsA Person) of Publication PartOf \( \sigma_{\text{Conference}=c} \) (Proceedings)
Context-Based Retrieval from the Web

Context capturing performed at the client side software
based on correlation-basic metrics for similarity
using advanced dictionaries, e.g., WordNet

Keyword extraction from the captured text and context
based on clustering algorithms

High-level classification of the query to a small set of predefined domains
applied to a set of search engines ranked by their relevance and coverage
depending on the keyword set

Ordering and adhesion of query results obtained from different search engines
by reranking with distance measures, adhesion and cohesion functions

variant of ‘blind’, non-informed search
may be enhanced by search algorithms, e.g. A*-algorithm
Indexing

- Library of media objects with their definition, update policy
- Library of dialogue scenes with their definition, usage profile, utilization history
- Advanced indexes of media objects (labels, search/retrieval, ordering) based on
  - resource identifier, citation identifier and location identifier (URI, URC, URL),
  - summaries for display of the main content of the media object,
  - docket for meta-information and processing information, and
  - annotations, ontology, classification of the media object.
- ...

• Library of media objects with their definition, update policy
• Library of dialogue scenes with their definition, usage profile, utilization history
• Advanced indexes of media objects (labels, search/retrieval, ordering) based on
  • resource identifier, citation identifier and location identifier (URI, URC, URL),
  • summaries for display of the main content of the media object,
  • docket for meta-information and processing information, and
  • annotations, ontology, classification of the media object.
Indexing

- Meta-information for classification, concept meaning, ontology, processing
  - Content provider, environment, security
  - Synonym, meronymy, homonymy, perspective and complementary antonyms, entailment (proper inclusion, backward presupposition, cause, troponymy)
  - Sources of classification, ontology, semantic interpretation, domain-dependent representation
  - Processing information, decomposition, display
Extraction of Summary

- Abstraction of the content with labels, names and specific structuring
  - naming, title, subject, identification, brief description (header)
  - body of the summary
    - structuring of the summary given as query form
    - layout applicable for representation
    - application options and areas, usability profile
    - renew policy and other processing information
  - additional information on authors etc.

- Utilization of database functions for extraction of views applicable to parameters of the query form
  - projection, selection, join, union, difference, intersection, comprehension
  - aggregation functions
    - distributive functions defined by structural recursion (min, max, sum, count)
    - algebraic functions defined by finite expressions of distributive functions (average, covariance, ratio)
    - holistic (statistics, rank, median, most frequent etc.) and other recursive functions
  - re-arrangement functions for ordering, grouping, weights etc.
Extraction of Summary

- ... 
- Computations based on functions of the host language
  - Additional information based on complex computations and/or provided by other sources
  - Generating the representation (fish-eye, cells of sheets, diagrams, graphics)
Dockets

Collins Dictionary of English Language, 1981

- piece of paper accompanying or referring to a package or other delivery, stating contents, delivery instructions, etc.
- summary of contents, as in a document
- list of things to be done
- sometimes serving as a receipt

Docket:

- Header information for dockets
  - Content providers, publisher, contributors and utilization information
  - Content reviewers and task specification (purpose of action, role of contribution)
  - Content evaluators and evaluation (interest in prescribing docket property types, statement of reviewers, groups interested in docket)
- ...
Dockets

Docket:

- ... 

- Applicability profile attached to the docket
  - trust and quality information, coverage, rights management, security modes
  - cooperation views and processes for various groups of users
  - applicable annotation and access models, resource type, format, language

- History of the media object
  - Versions, releases and their utilization profile
  - Update information, link integrity checking, and binding style
  - Processing signatures, docket traces, seals of approval, usage indicators
Extending to Interaction Pattern

Why not style guides?
- Too simple and abstract
- Difficult to select
- Difficult to interpret
- Can be conflictive
- Style guides are of the form: Do this or Do not do this.

Patterns?
- Should present and solve a problem.
- Should have a context of use where the solution is reachable.
- Should be useful in the treatment of similar problems in other situations.
- Should teach about the problem solution.
- Should have an identification name to identify the pattern.
- Nevertheless, patterns are no as restrictive as style guides.
Interaction Pattern

I. Ramos: “The difference between design patterns and interaction patterns is in their application domain. The first one are intended for object domain implementation (OO software design) whereas interaction pattern are intended for user interface objects (GUI objects). The first one aims reuse and the second one aims usability.”

- Replace style guides with interaction patterns. Associate pseudo-idioms to the defined interaction patterns using XML based languages.
- In parallel analysis, design and implementation.
- Use of declarative models to describe the different aspects of the user-system interaction.
- Automatic code generation starting from the models.
- Formal and methodological support.
- Usability criteria within the development process.
- Portability to different devices.
User Interface Models

- Task model (objective, actions, artifacts)
  - ordered set of activities and actions the user has to perform to achieve a particular purpose or goal
  - artifact/scene: essential for task development (modeled by media objects)

- User model
  - Purpose: creation of individual and personalized user interfaces
    - characteristics of different types of users
    - specification of tasks the actor can perform (depending on the history, profile, role etc.)
  - Adaptation features
    - adapted user interface: different roles of a user (supervisor, administrator, employee, ...)
    - adaptive user interfaces: depends on the actor profile (child, handicapped, ...)
  - User group profile for representation of specific properties of the user

- Dialogue scene model
  - “Conversation” between user and system
  - Follows an event-process paradigm (syntactic structure of the user system interaction)
  - Presentation models (screen characteristics, dependencies among them)
SiteLang

Story space - space of all specified stories

Story - labeled graph of different integrated mini-stories

Scenario - run through a system by a cooperating set of actors

Scene of a mini-story - plot

Dialogue scene - conditional actions of an enabled actor on page based on provided media objects

\[ ru_i: \text{on event if precond do actions accept on postcond} \]

- if precond\text{r}_u_i \text{ and event then actions and CommitState}_r_u_i = \text{toCommit} \\
- if CommitState_r_u_i = \text{toCommit and postcond}_r_u_i \text{ then Commit}_r_u_i \\
- if CommitState_r_u_i = \text{toCommit and } \neg \text{postcond}_r_u_i \text{ then Undr}_u_i \\

Interaction space - space of all possible interaction stories

System space - glass box of system (all runs integrated)
The General Picture

\[(\text{enabled} \cdot \text{used} \cdot \text{DialogueExpression}) \Rightarrow \text{uses}[\text{Scene}] \subseteq \text{enabled} \cdot \text{involved}[\text{Scene}]\]

1. **Story**
   - in **Activity Sequence**

2. **Dialogue Expression**
   - uses **Scene**
     - (1,1)

3. **used**
   - **enabled**
     - **involved**

4. **Dialogue Step**
   - **Context**
     - **Actor**
       - **Task**

- **Context**
  - **Equipment**

- **Actor**
  - **Group**
  - **Profile**

- **Representation Style**
  - **Media Object**
    - **Right Object**
      - **Right Category**

- **Dialogue Step**
  - **Condition**
    - **Do**
    - **AcceptCond**

- **Information**
  - **Content**
    - **Topic**

- **Event**
  - **Emphasis**
    - **Default**

- **Usage**
  - **Right**
    - **Obligat**

- **Task Assignment**
  - **Role Category**

- **Scene**
  - **BasedOn**
    - (1,1) \(\vee (1,n)\)

- **Particular**
  - **Channel**
**Interaction Modeling (1)**

*Story* as labeled di-graph $S = (V, E, \lambda, \kappa)$

- $V$ - scenes, $E \subseteq V \times V$ - transitions
- media object assignment $\lambda : V \rightarrow \{\text{MediaObj}\}$
  - representation through media objects
  - with permitted rights, permitted roles, obligations of actors
- MediaObject (Container, ManipulatRequests, SuppliedFunctions)
- activity assignment $\kappa : E \rightarrow \{\text{Activity}\}$
  - activity = ( actor(profile) , task,
    - context (equipment, channel, rights, roles, particular),
    - representation (style, default, emphasis, ...))

*Scenario* - run through the system

- with cumulative history (and adaptation)
- consists of scenes
  - visited sequentially or in parallel by actors
- story space - composition of sequences
Media Objects and Containers

- (abstract) container
- (abstract) information unit
- raw information unit
- structure
- operations
- view
- interface
- content tailoring
- usage
- content
- abstract presentation
- context
- supplied DBMS functions
- DB manipulation requests
- media object

Overview
Refinement
Mapping
Background
SiteLang

Adaptation
16.10.2014
B. Thalheim
Scene Specification

**Scene expression** $v \in \text{Alg}(\text{DialogueStep})$

- basic control: sequence `;`, parallel split `|`, exclusive choice `|`, synchronization `\text{sync}`, simple merge `+`

- advanced branching and synchronization control: multiple choice, multiple merge, discriminating, n-out-of-m join, synchronizing join

- structural control: arbitrary cycles `*`, implicit termination `\downarrow`

- control on multiple objects: CMO with a priori known design time knowledge, CMO with a priori known runtime knowledge, CMO with no a priori runtime knowledge; CMO requiring synchronization (synchronization edges)

- state-based control: deferred choice, interleaved parallel executing, milestone

- cancellation control: cancel action, cancel case

**FRAME:**

```
on event if \alpha doScene v accept on \gamma
```

representation of scenes via Montages

control and object flow specification
Interaction Modeling and Specification 3

Dialogue scene

on event if α then actions accept on β

```
dialogueObject ::= expression

if state condition ∧ event condition ∧ actor condition

dialDo: expression

dialIf: if acceptCondValue = true then CurrentTask = T
else CurrentTask = undo endif
```
Story and Side Stories

(1) story: many-dimensional (multi-layered) graph

Lecture scheduling system: dimensions for chairs and their members, for faculty staff, organizing stuff, students and assistants

side story changing the story state partially

main story (for example as a sequence)
scenario (run through story space) can or cannot use side paths

Example: Tootsie has 5 interleaved (sub-)stories which are supporting story development in different stages

(2) generic scene: folded by opportunities

Lecture scheduling system: Planning scene as views with different visibility of data (dean’s team - full; chairs - restricted view; program chair - full for program)
Representation of Dialogue Steps within a Dialogue Scene

- dialogue scene
  - control(event, preCondition, acceptCondition)
  - sub-unit supplied process
  - enabled actor
  - transition according to dialogue scene expression

- dialogue step

involved actors
story scene sequence
media object
representation style
context, task
Dialogue Steps for Event etc. Search

- event search scene
- map browsing step
- points of interest
- individual request step
- entry step
- target seeking step
- property-based search
- result & clarification step
- booking step
Specific Media Types: Work Place

Overview
Refinement
Mapping
Background
SiteLang

Information
Content
Topic
Concept
Specific Media Types: Work Room
Specific Media Types: Communication Place
Specific Media Types: Archive

Overview
Refinement
Mapping
Background
SiteLang

Content
Information
Concept
Topic
Specific Media Types: Market Place