Analysis

- Study of a system (e.g., a software system to be built)
- Objectives
  - understand system
  - document understanding
  - promote a common understanding among participants (e.g., clients, analysts, developers, users, managers)
- Key activities
  - studying
  - learning
  - writing
  - communicating

Analysis of Customer Requirements

There’s more than just systems analysis.

- Understand application context
  - general business objectives
  - specific business needs to be addressed by the system
- Understand business constraints
  - cost
  - schedules
  - legal considerations
- Understand the technical constraints
  - available technical resources (hardware, software, personnel)
  - technology and expertise that can be brought in for the project

Needs and Feasibility Assessment

- Suggested Alternative
- Primary Alternative
- Feasibility Consideration
- Economic Consideration
- Legal Consideration
- Constraint
- Schedule Constraint
- Need
- Cost Constraint
- Technical Consideration
- Goal
- Trade-off
- Criteria

Using OSM for Needs and Feasibility Modeling

- Standard guide to assessing needs and feasibility
- Standard, but can be tailored for particular projects
- Can be prepopulated with data and used for several projects
- Same model for assessment and application development
- Can query the assessment model with standard query languages (in the same way as the application model)

Query Languages for OSM

- OSM-QL
- OSM-Algebra
- OSM-Calculus
- OSM-SQL
Chapter 7 - OSM-Algebra

List goals for the primary alternative.

\[ \text{Goal} \rightarrow \text{Suggested Alternative} = \text{Primary Alternative} \]

List suggested alternatives that satisfy all needs.

\[ \text{Goal} \rightarrow \text{Need} = \text{Goal} \rightarrow \text{Suggested Alternative} \Rightarrow \text{Primary Alternative} \]

Chapter 7 - OSM-Calculus

List goals for the primary alternative.

\[ \{ <x> | \exists y (\text{Goal}(x) \land \text{Suggested Alternative}(y) \land \text{Primary Alternative}(y)) \} \]

List suggested alternatives that satisfy all needs.

\[ \{ <x> | \not\exists y (\text{Goal}(x) \land \text{Suggested Alternative}(y) \land \text{Primary Alternative}(y)) \} \]

Chapter 7 - OSM-SQL

List goals for the primary alternative.

\[ \text{select Goal} \]
\[ \text{from} \text{ satisfies, Goal, Primary Alternative} \]
\[ \text{where} \text{ Suggested Alternative} = \text{Primary Alternative} \land \text{Goal} = \text{Need} \]

List suggested alternatives that satisfy all needs.

\[ \text{select Suggested Alternative} \]
\[ \text{from} \text{ Suggested Alternative A} \]
\[ \text{where not exists } <x> \]
\[ \text{select } <x> \]
\[ \text{from} \text{ Need N} \]
\[ \text{where not exists } <x> \]
\[ \text{select } <x> \]
\[ \text{from} \text{ satisfies S} \]
\[ \text{where N.Need} = \text{S.Need} \land \text{A.Suggested Alternative} = \text{S.Suggested Alternative} \]

Chapter 7 - Analysis Methods

- Analysis is a knowledge-discovery task
  - learning is never easy
  - requires deep involvement with information
  - often faced with vague and contradictory information
  - experience counts

- General approaches
  - ask questions, observe activities, read documents and forms
  - learn by formal writing and by model construction

- Specific approaches
  - methodologists: e.g., scenarios, use cases, CRC cards, ...
  - proprietary approaches often used in consulting firms
  - model-driven approach (OSM)

Chapter 7 - OSM Model-Driven Analysis

- Ontological model
  - models can either expand or limit our ability to analyze
  - best: models that let us directly record structural and behavioral concepts as found in the system being modeled

- Match analysis approach with application characteristics
  - ORM approach
    - data intensive application components
    - examples: personal records, business data, scientific data
  - OBM approach
    - components with intensive individual object behavior
    - examples: behavior drivers for elevators, machining, traffic lights
  - OIM approach
    - components with intensive interactive behavior
    - examples: automatic-teller machines, airline reservations, sensors

Chapter 7 - Application-Model Integration

- Large projects
  - several analysts each tackle a manageable-size component
  - systematically integrate these components

- Approach to integration
  - integration framework and strategy
    - framework: high-level OSM application model
    - strategy: decisions about how to proceed and how to resolve conflicts
  - integration activities
    - compare: identify correspondences and conflicts
    - conform: resolve conflicts
    - merge: put components together and reorganize as warranted
Chapter 7 - 13

Example: Integration Framework

```
Room is reserved for Guest
```

High-level diagrams work well for specifying integration frameworks.

Chapter 7 - 14

Example: ORM Integration

Name conflict: Name of Room vs. Name of Guest; Person vs. Guest
Structural conflict: reservations
Constraint conflict: minimum participation of Room in reservations

Chapter 7 - 15

Example: Integrated ORM

```
Room NrDaysArrivalDate
Guest
Future
Guest
Current
Guest
City
StreetNr
Name
Room
GuestNr
Cost
RoomName
RoomNr
1 has 1
1 has 1 has 1:*
1 has 1
1:* 1:*
1:*
1:*
1:*
1:*
1:
1:*
1:*
1:*
1:*
0:* 1
0:*
0:*
Name conflict: Name of Room vs. Name of Guest; Person vs. Guest
Structural conflict: reservations
Constraint conflict: minimum participation of Room in reservations
```

Chapter 7 - 16

Example: OBM Integration

```
Reservation Clerk @ request guest list
list arriving guests
Exists @ special guest arriving and not yet notified proprietor
notify proprietor
@ cancel
cancel reservation
Ready @ new reservation
make reservation
Reservation Clerk @ request guest list
list arriving guests
Exists @ special guest arriving and not yet notified proprietor
notify proprietor
@ cancel
cancel reservation
@ new reservation
make reservation
```

Chapter 7 - 17

Example: Integrated OBM

```
Reservation Clerk
@ request guest list
list arriving guests
Ready @ special guest arriving and not yet notified proprietor
notify proprietor
@ cancel
cancel reservation
@ new reservation
make reservation
```

Chapter 7 - 18

Analysis Validation

- Checking a model for completeness and correctness
  - Discovery (Chapter 7)
  - Formalization (Chapter 8)
  - Requires skill and insight ("Ya gotta use your head.")
- Discovery techniques
  - Formal technical review
  - Systematic scrutiny based on queries, behavior scenarios, and interaction sequences
  - "White-box and black-box testing"
  - Object classification
    - Cohesive sets of objects
    - Structural similarity
    - Behavioral similarity
White-Box Validation

- “Fire” each transition
- “Execute” each state net
- Check correctness
- Check completeness
  - Are all expected/required scenarios addressed?
  - Are all exceptions handled?

Black-Box Validation

- ORM perspective
  - Pose and “execute” ad hoc queries.
  - Does the application model contain the information necessary to answer these queries?
- OIM perspective
  - Consider stand-alone interactions and interaction sequences.
  - Does the application model include all interactions and interaction sequences required in the system being built?
- OBM perspective
  - Check the receiving end of interactions.
  - Are all required application-model interactions handled?

Object Classification

- Why classify?
  - may discover new object sets
  - may cause us to better organize object structure and behavior
  - may help us better understand an analysis application model
- Classify based on similarity
  - the more two objects are related to other objects in the same way, the more they are similar
  - the more two objects behave the same, the more they are similar

Congruency

- An object set is congruent if the common properties of the objects in an object set $S$ coincide with the properties explicitly defined for $S$.
- Explicit properties – given in application model
- Common properties
  - Common relationship-set property: all objects participate
  - Common state-net property: all objects can be in state or transition
  - Common interaction property: all objects can interact the same
- Incongruent:
  - Overstatement: explicit property not common to all
  - Understatement: common to all but not explicit

Structural Congruency

<table>
<thead>
<tr>
<th>Congruent:</th>
<th>Incongruent:</th>
</tr>
</thead>
</table>

| Congruent: |

Behavioral Congruency

Incongruent Congruent
Chapter 7 - 25

**Congruency and Integration**

- **ReservedRoom**
- **OccupiedRoom**

**Congruent but not integrated**

Chapter 7 - 26

**Integrated But Not Congruent**

- **Missing Generalization** (Room)
- **Missing Specializations** (CurrentGuest, FutureGuest)

Chapter 7 - 27

**Overstatement**

- ** Explicitly Defined but not Common**

Chapter 7 - 28

**Overstatement Resolution**

- **Add Missing Specializations**

Chapter 7 - 29

**Understatement**

- **Common but not Explicit**

Chapter 7 - 30

**Understatement Resolution**

- **Move Explicit Property to Generalization**
Integrated and Congruent