Requirements Engineering for Sustainability

Birgit Penzenstadler
birgit.penzenstadler@csulb.edu
www.csulb.edu/~bpenzens

@twinkleflip
#SustainabilityDesign
#KarlskronaManifesto

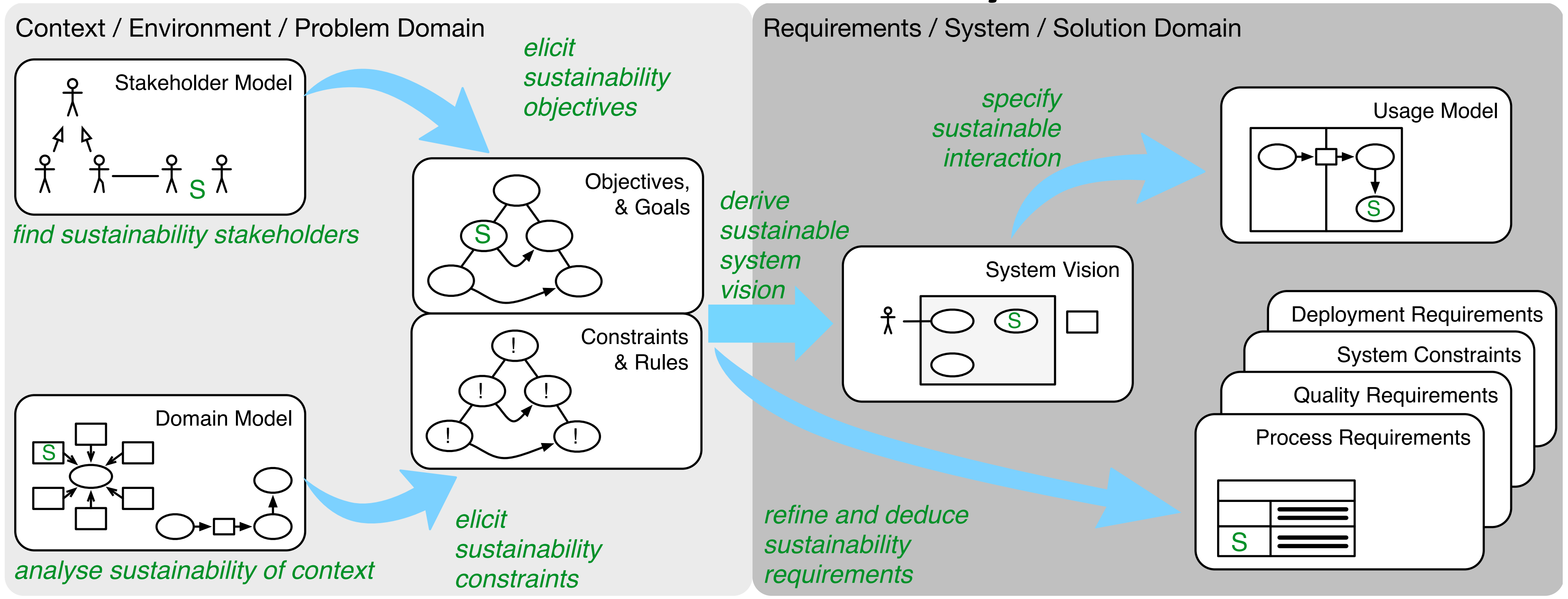
Timeline

- Tuesday 29.3
  - 10-12 Open lecture “Software engineering for sustainability – The Karlskrona manifesto”, Room 4511
  - 12-16 Opening of the course, Room 7441
- Wednesday 30.3
  - 18-22 LUT Beach Sauna, student idea presentations & discussions
- Thursday 31.3
  - 10-12 Stakeholder model and goal modelling, Room 4511
  - 12-14 Course work, Room 4511
- Friday 1.4
  - 10-12 System vision, Sustainability analysis and use cases, Room LS204
  - 12-14Course work, Room LS 204
- Monday 4.4.
  - 10-14 Intermediate presentations, Room 7441
- Tuesday 5.4
  - 12-16 Course work, Room 7441
- Wednesday 6.4
  - 8-10 Briefing for presentations, Room 7441
  - 10-12 Course work, Room 7441
- Thursday 7.4
  - 10-14 Course work, Room 7441
- Friday 8.4
  - 12-16 Final presentations, Room 7441

Outline & Overview

1. System Vision
2. Usage Model

Requirements Engineering for Sustainability


Context / Environment / Problem Domain
Stakeholder Model
S
find sustainability stakeholders
elicit sustainability objectives
Objectives, & Goals
S
Constraints & Rules
!
Domain Model
S
analyse sustainability of context
elicit sustainability constraints
Requirements / System / Solution Domain
derive sustainable system vision
System Vision
S
specify sustainable interaction
Usage Model
S
Deployment Requirements
System Constraints
Quality Requirements
Process Requirements
S
refine and deduce sustainability requirements

Example checklist for analyzing environmental sustainability for a software system.

Guiding Questions for Green RE:

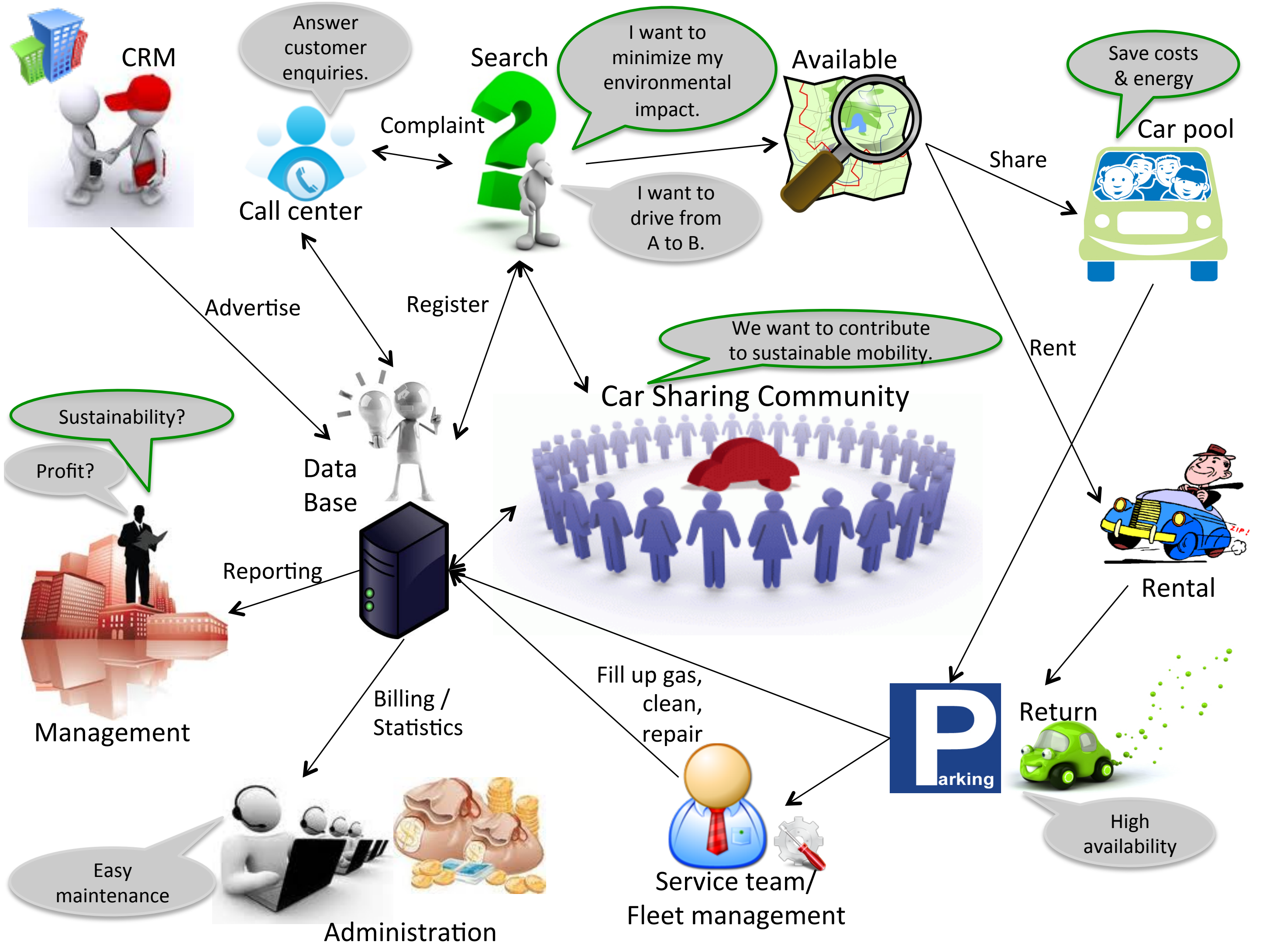
1. Does the system have an explicit sustainability purpose?
2. Which impact does the system have on the environment?
3. Is there a stakeholder for environmental sustainability?
4. What are the sustainability goals and constraints for the system?

System Vision

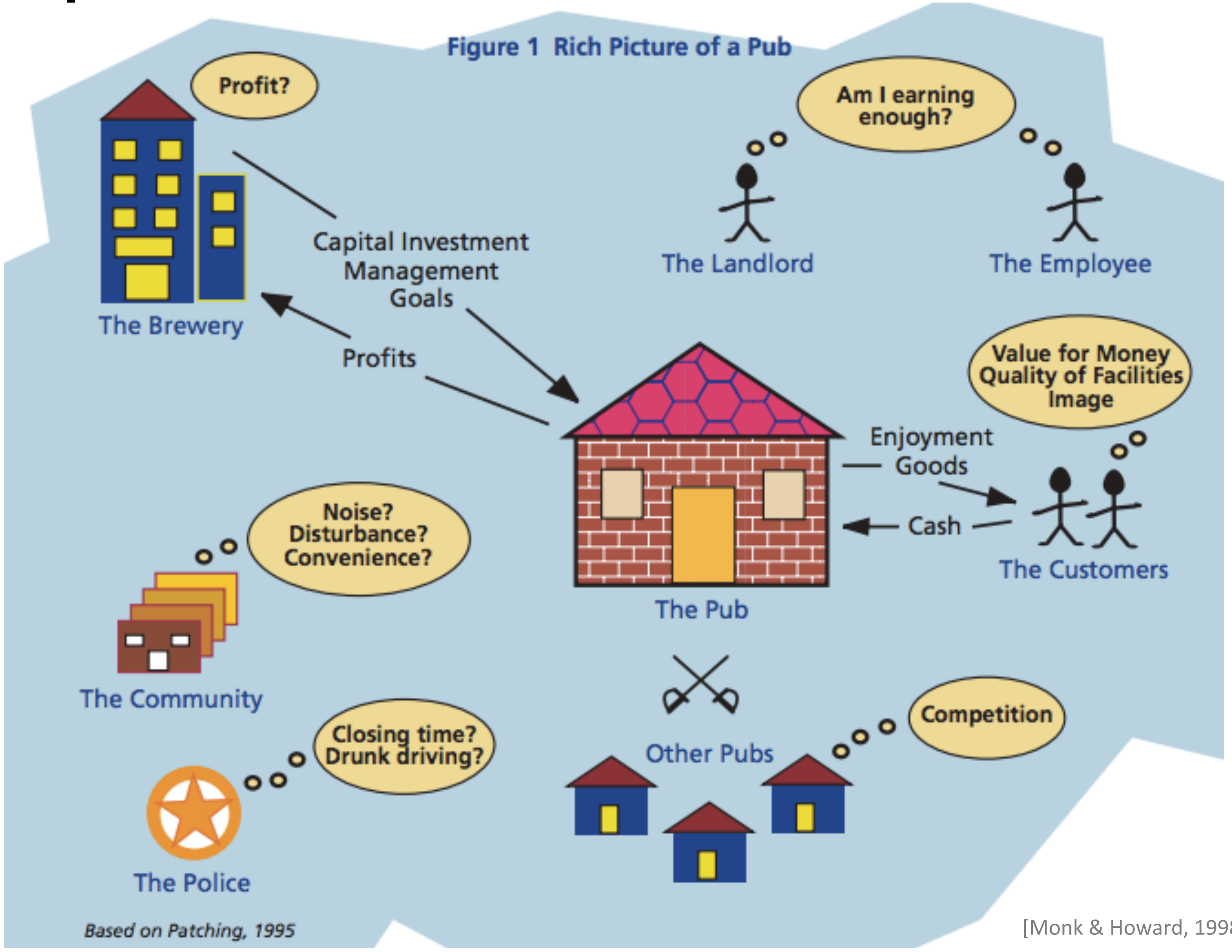
Definition: System vision

- Def.: The system vision is a joint vision of the system agreed upon by all active stakeholders
- Characteristics
  - Big picture
  - Abstract
- Purpose
  - Agreement on *what this project is about*
  - Easy communication with stakeholders

Example: Car Sharing System



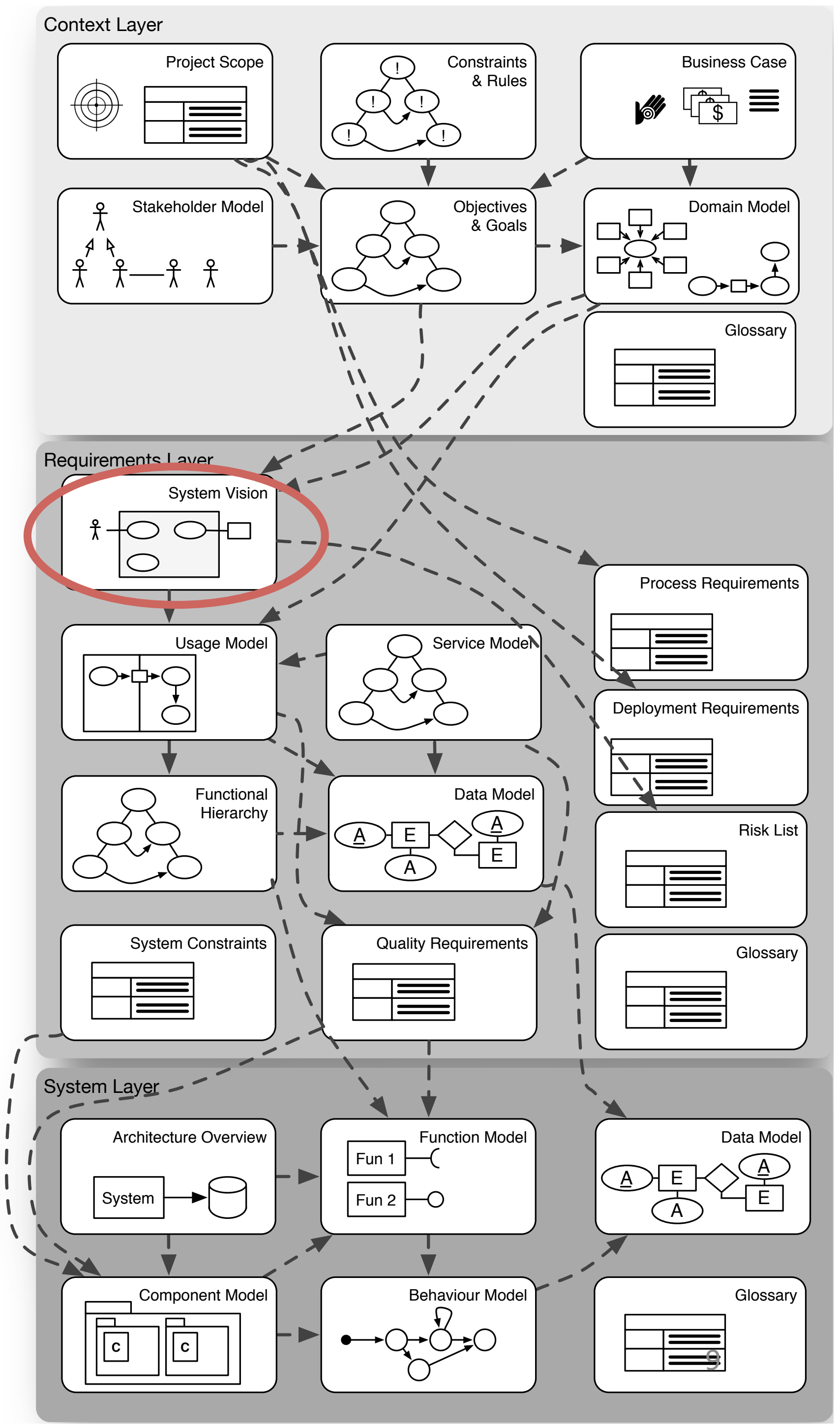
Example: Pub



[Monk & Howard, 1998]

Connection to RE content items

- **Input**
  - Business Case
  - Stakeholders
  - Goals
  - Domain Model
- **Output**
  - Usage Model
  - Quality Requirements
  - Risk List



Methods

- Rich Picture
  [Monk & Howard, 1998]
- Used in participatory design
  - Brainstorming
  - Storyboarding
  - Paper-based prototyping

Method: Rich picture

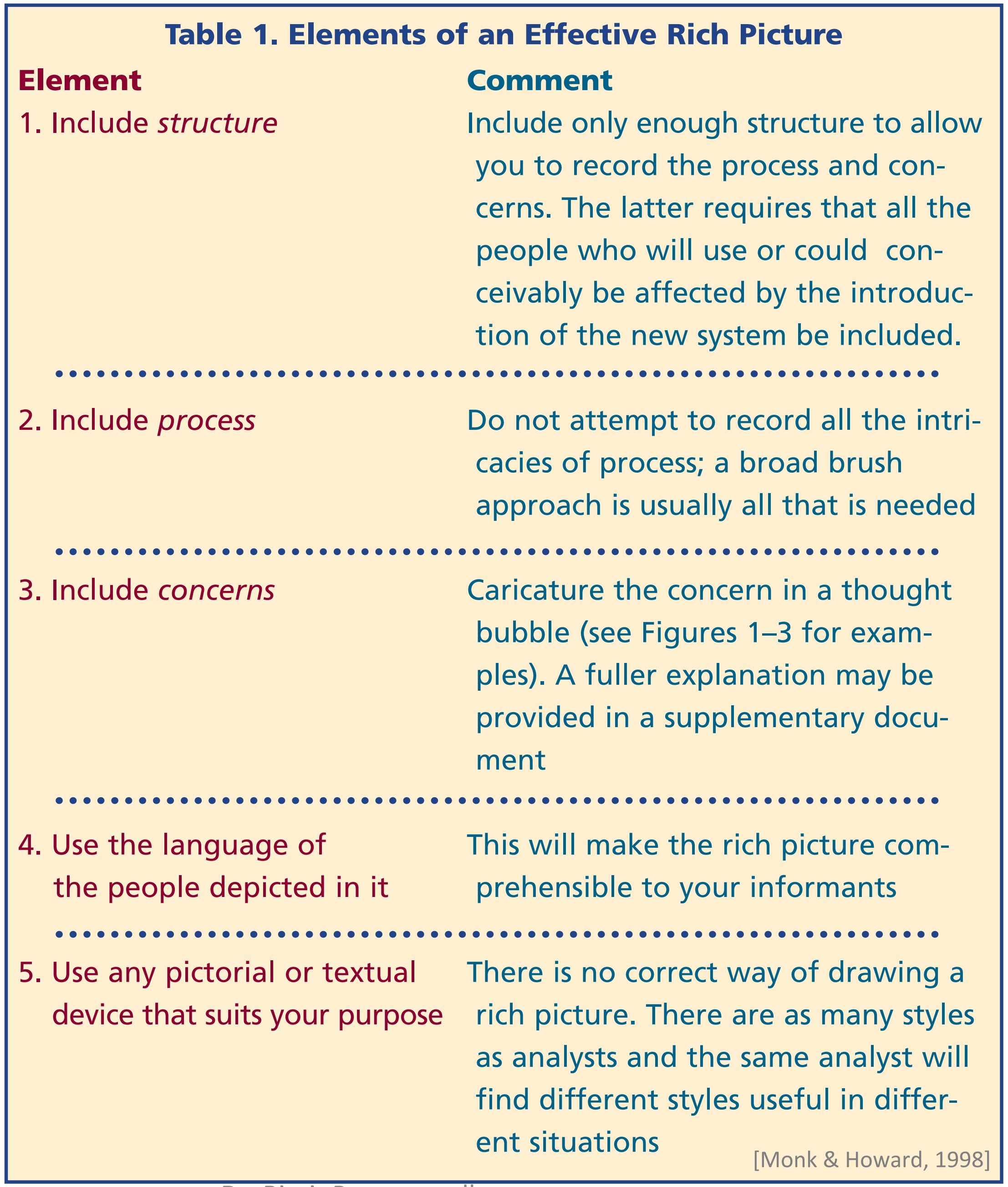


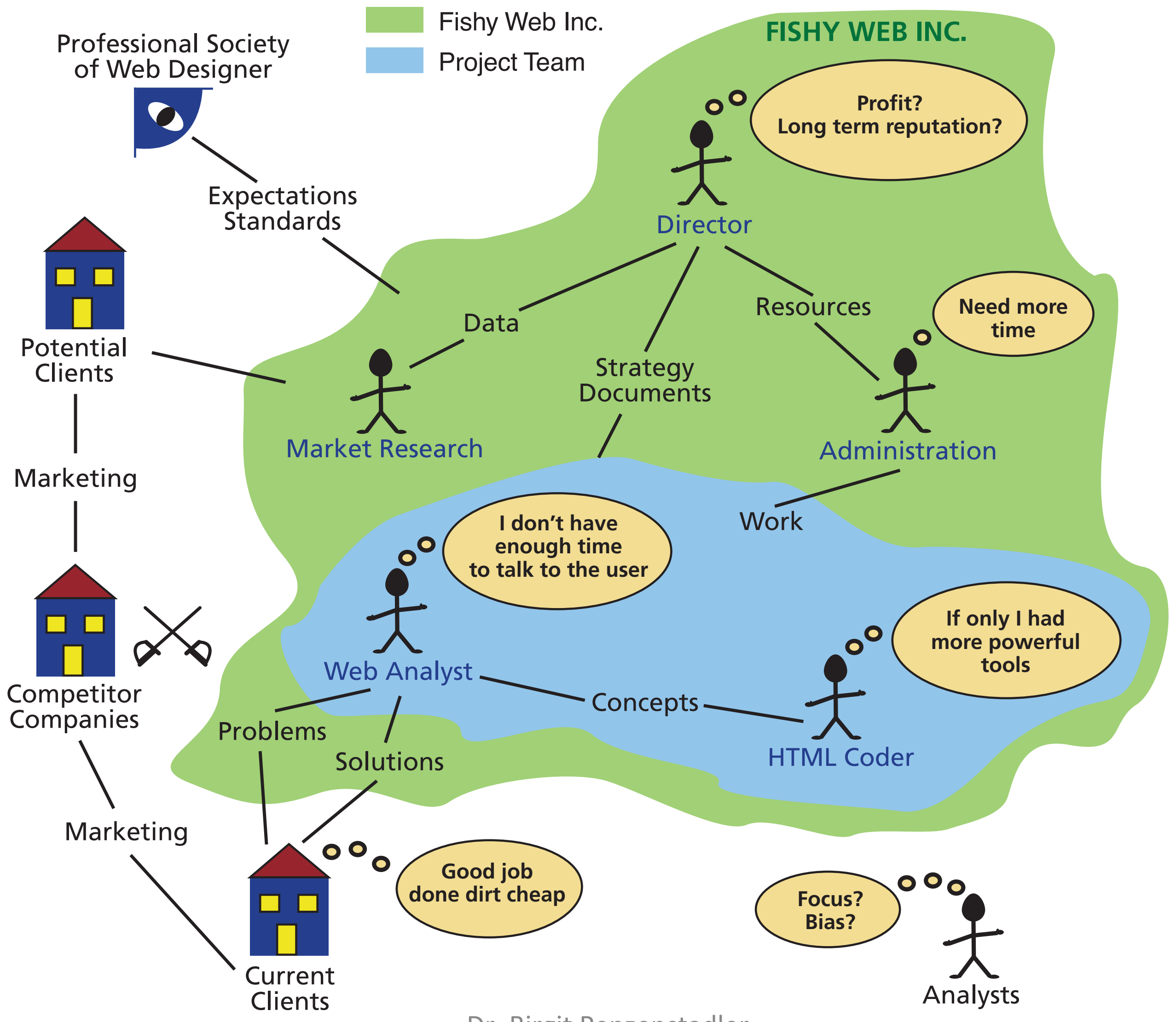
Table 1. Elements of an Effective Rich Picture

Element	Comment
1. Include *structure*	Include only enough structure to allow you to record the process and concerns. The latter requires that all the people who will use or could conceivably be affected by the introduction of the new system be included.
2. Include *process*	Do not attempt to record all the intricacies of process; a broad brush approach is usually all that is needed
3. Include *concerns*	Caricature the concern in a thought bubble (see Figures 1–3 for examples). A fuller explanation may be provided in a supplementary document
4. Use the language of the people depicted in it	This will make the rich picture comprehensible to your informants
5. Use any pictorial or textual device that suits your purpose	There is no correct way of drawing a rich picture. There are as many styles as analysts and the same analyst will find different styles useful in different situations

[Monk & Howard, 1998]

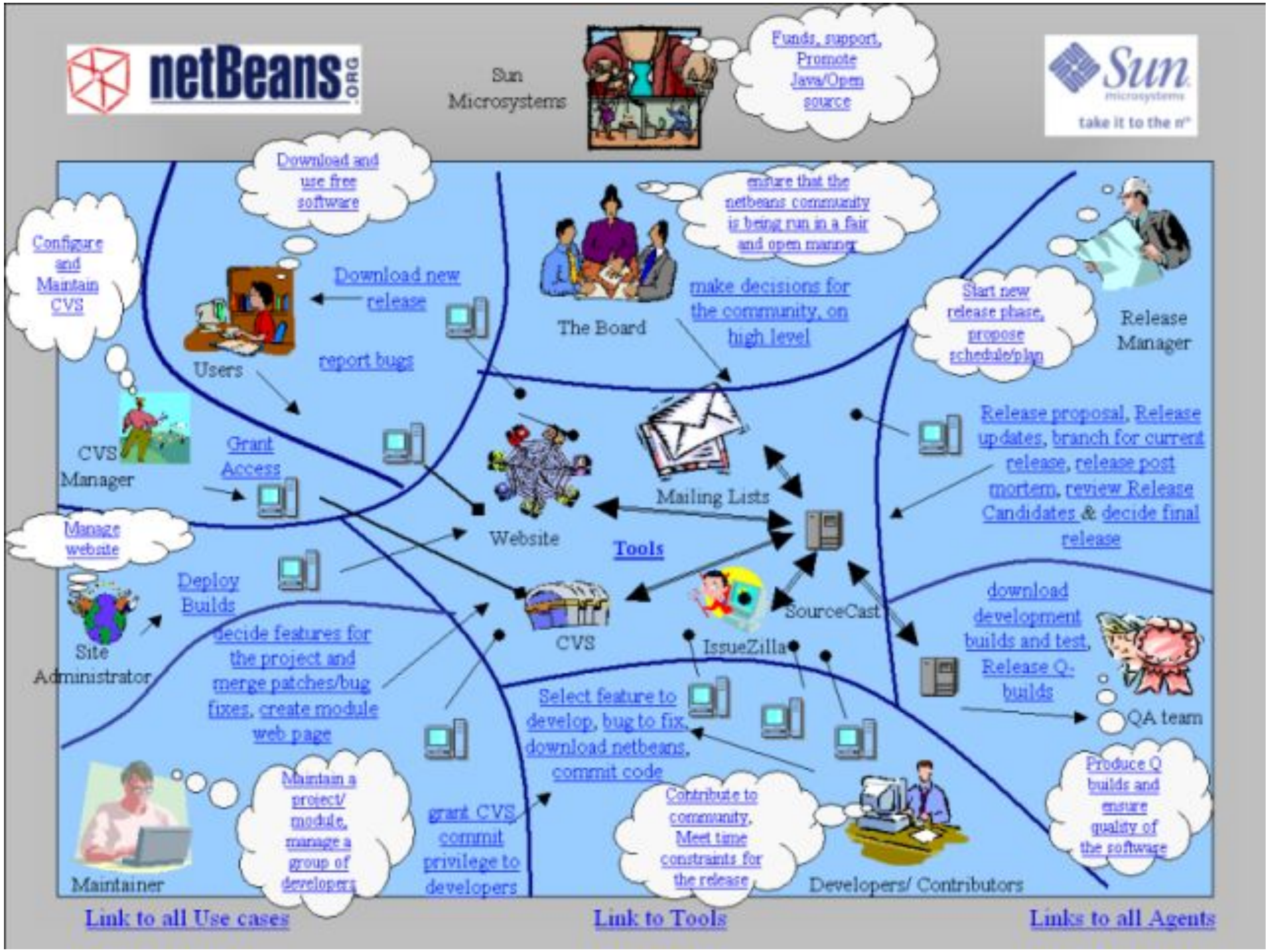
Example: Web Design Consultancy

Figure 2 Rich Picture of Web Design Consultancy



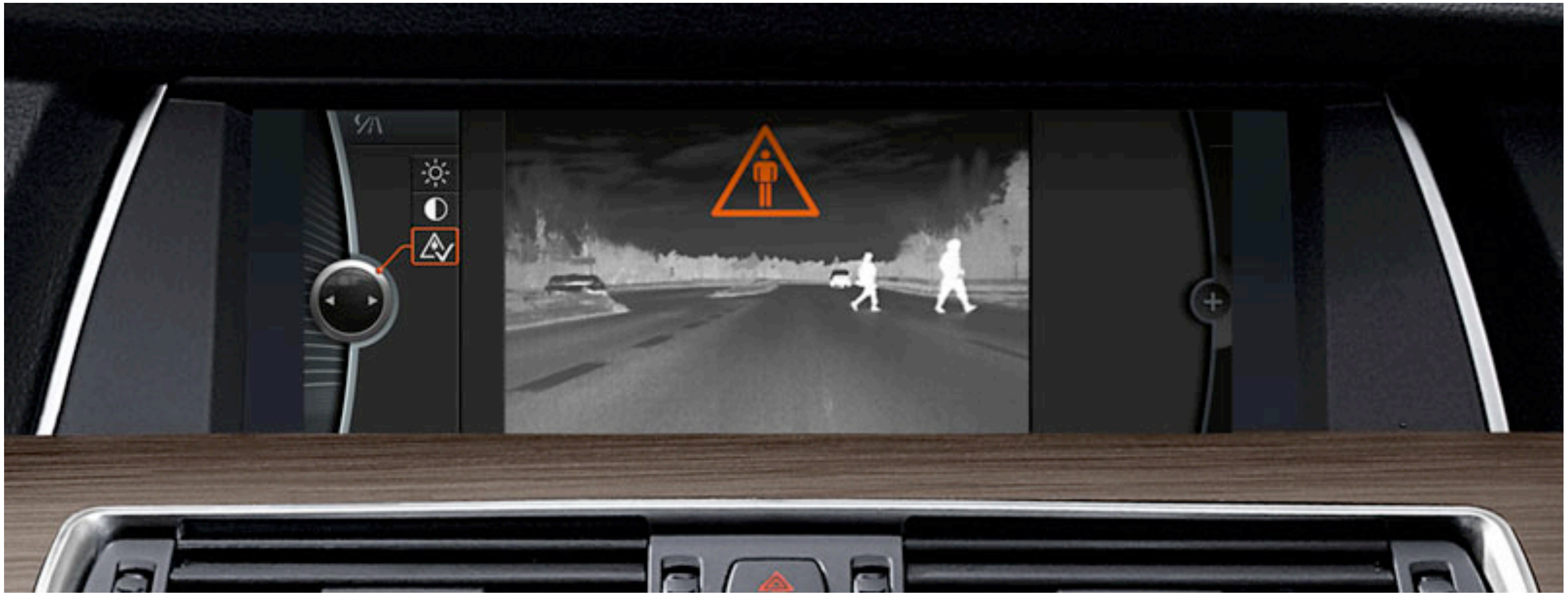
[Monk & Howard, 1998]

Example/exercise: What is this system?


netBeans.ORG
Sun Microsystems
Funds, support, Promote Java/Open source
Sun microsystems take it to the nth
Download and use free software
Configure and Maintain CVS
Download new release
Users
report bugs
ensure that the netbeans community is being run in a fair and open manner
The Board
make decisions for the community, on high level
Start new release phase, propose schedule/plan
Release Manager
CVS Manager
Grant Access
Website
Mailing Lists
Release proposal, Release updates, branch for current release, release post mortem, review Release Candidates & decide final release
Manage website
Deploy Builds
Tools
Site Administrator
CVS
SourceCast
IssueZilla
download development builds and test, Release Q-builds
QA team
decide features for the project and merge patches/bug fixes, create module web page
Select feature to develop, bug to fix, download netbeans, commit code
Maintainer
Maintain a project/ module, manage a group of developers
grant CVS commit privilege to developers
Contribute to community, Meet time constraints for the release
Developers/ Contributors
Produce Q builds and ensure quality of the software
Link to all Use cases
Link to Tools
Links to all Agents

UC Irvine, Oza et al. 2002

Example/exercise: What is this system?



UC Irvine, Oza et al. 2002 14

Example: Cold Storage Warehouse

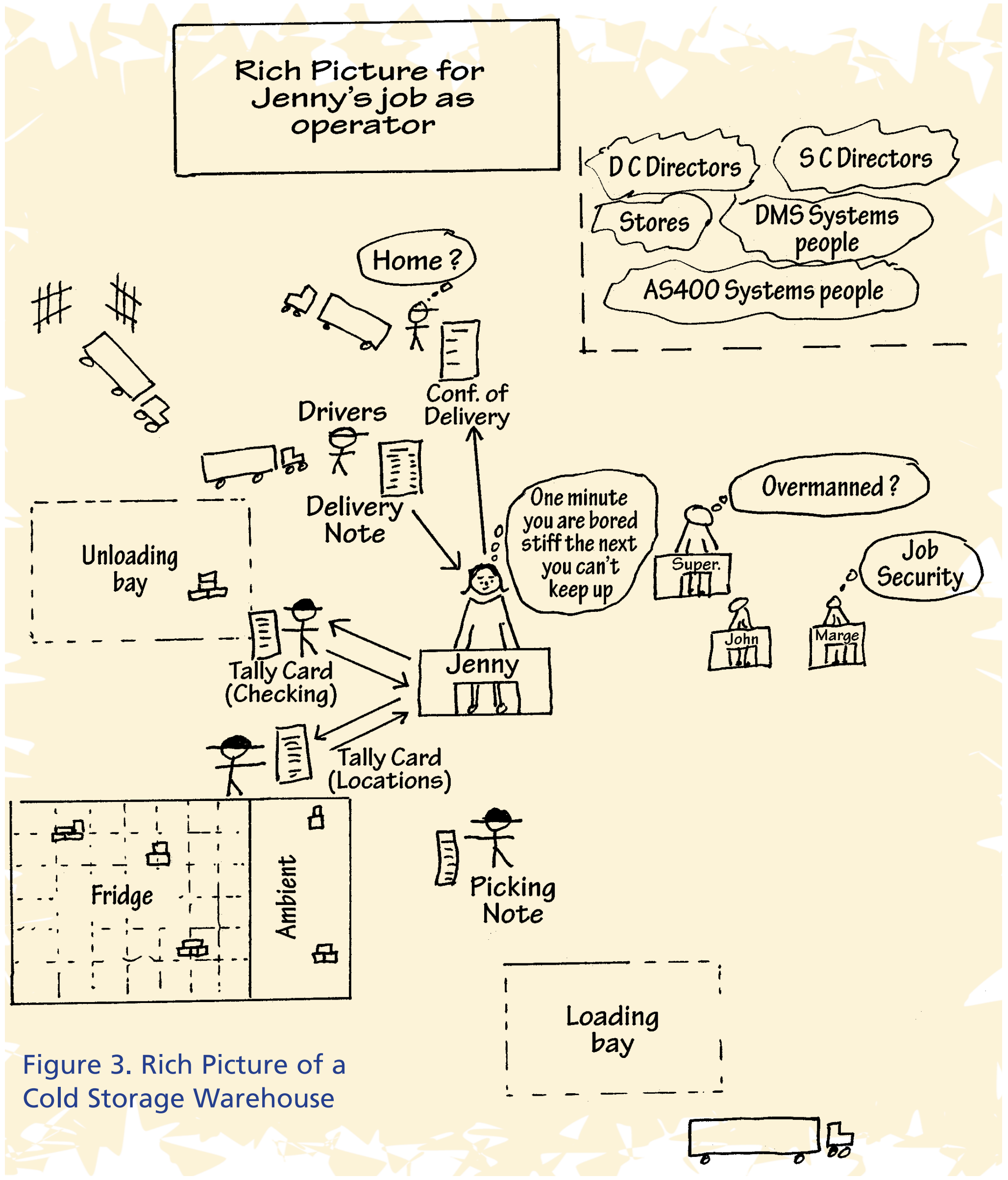
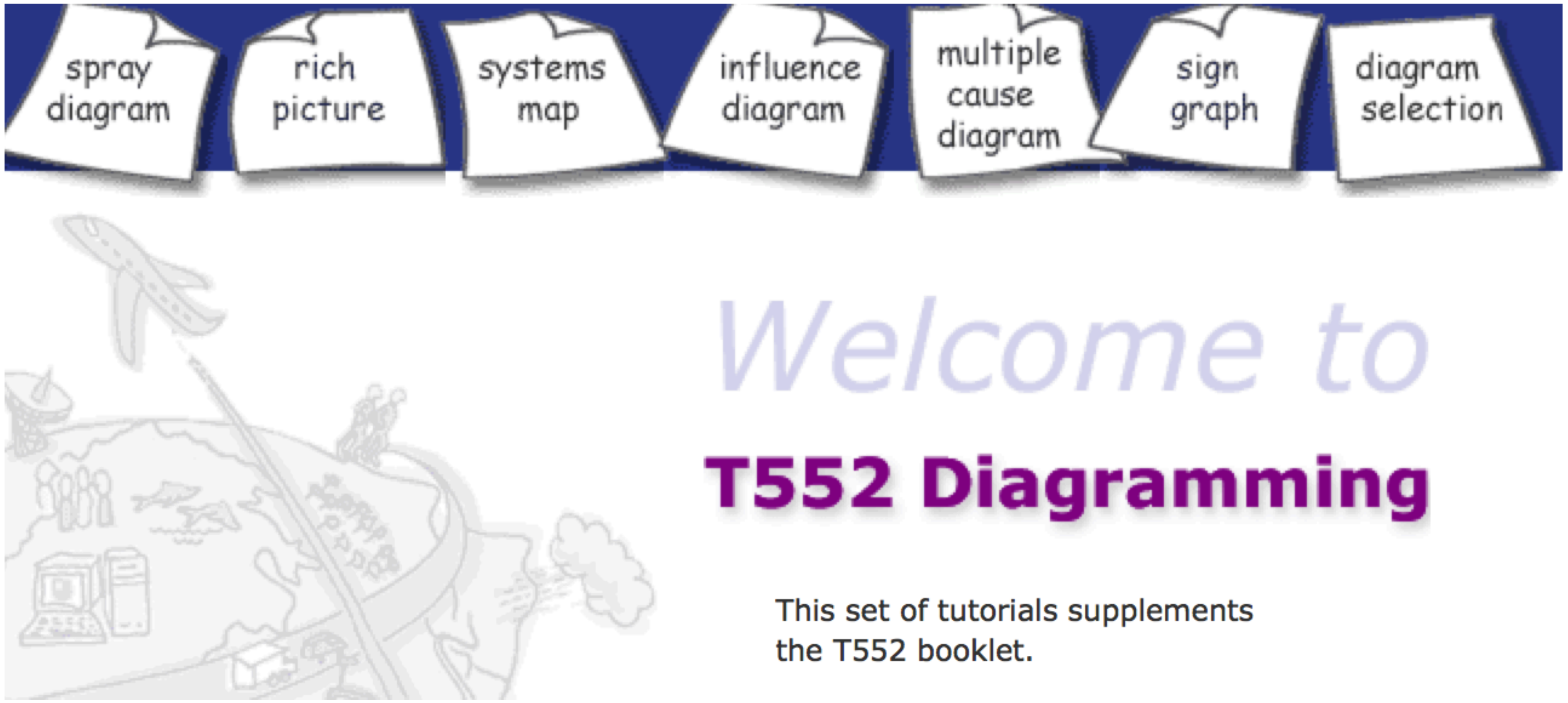


Figure 3. Rich Picture of a Cold Storage Warehouse

[Monk & Howard, 1998]

Good tutorial

http://systems.open.ac.uk/materials/T552/



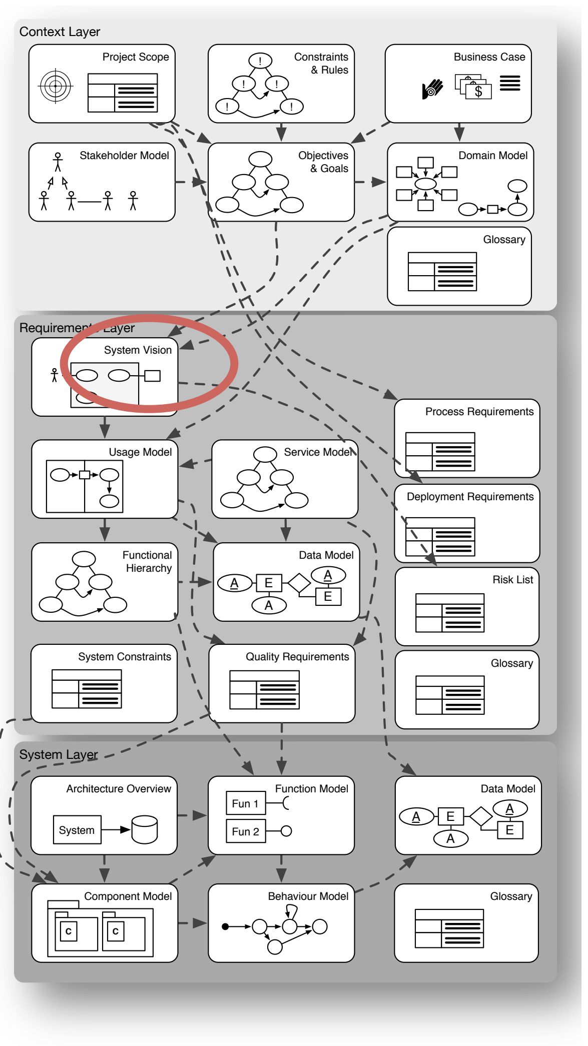
Welcome to

T552 Diagramming

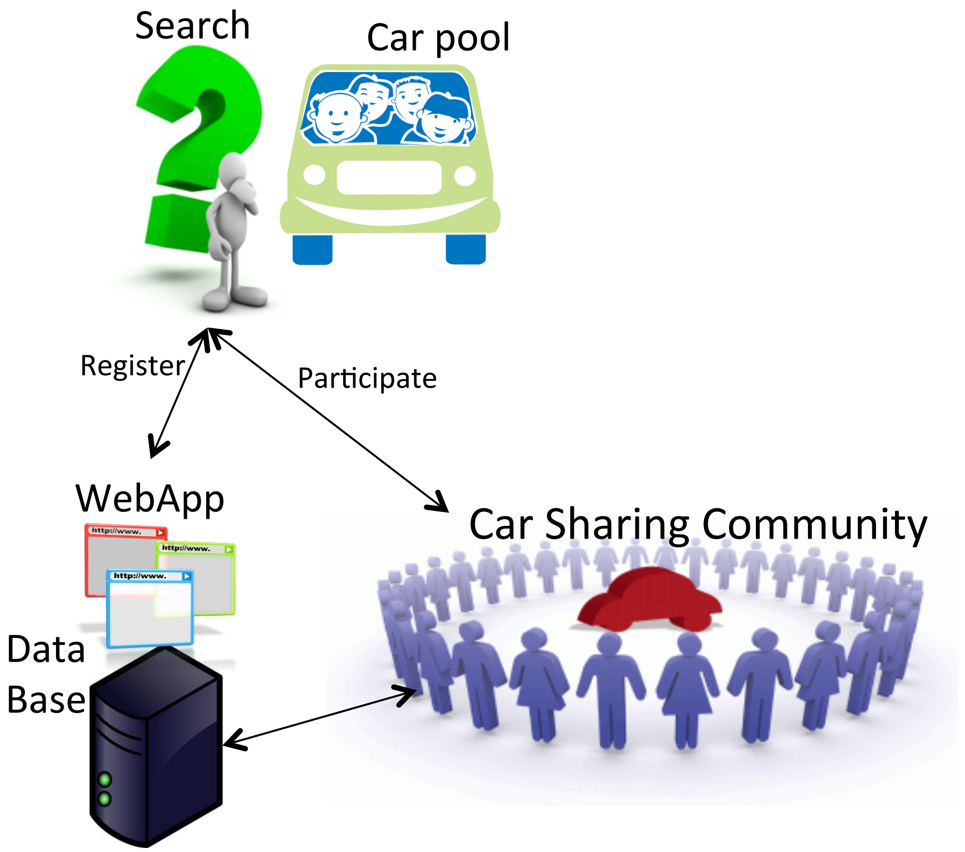
This set of tutorials supplements the T552 booklet.

System Vision in AMDiRE

- Includes
  - Structure
  - Process
  - Concerns
- Elements
  - System border
  - Others systems in the context
  - Features / usage
  - Relation to important stakeholders

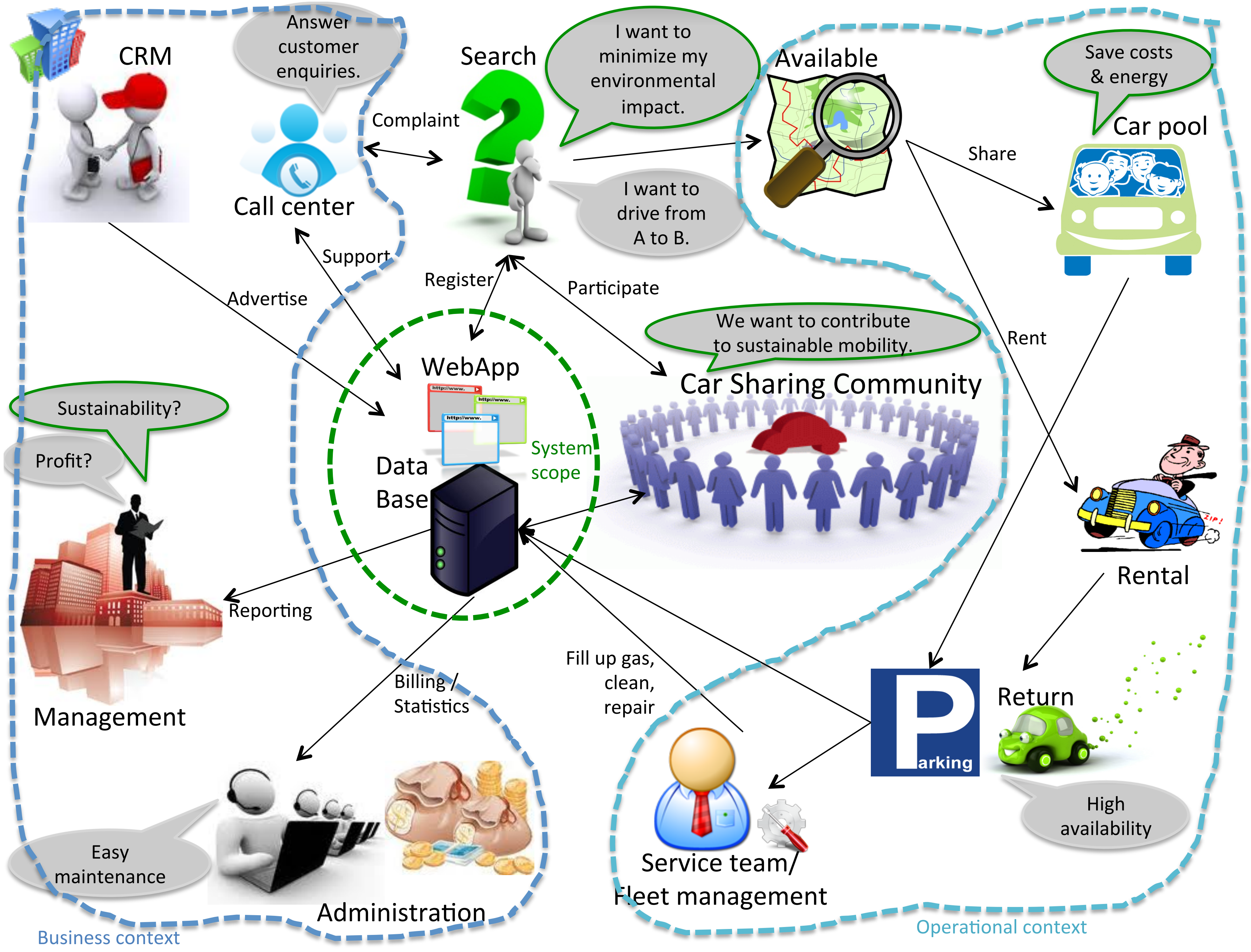


System Vision



→ How to:
Take input from Stakeholder Model, Domain and Goals to sketch:

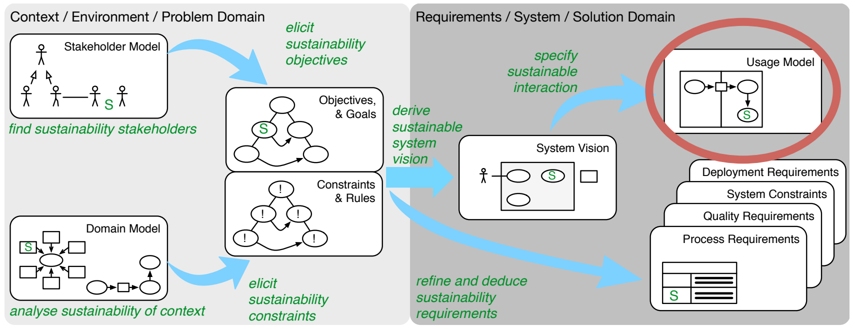
- System scope
- Major features
- Business context
- Operational context
- Stakeholders
- Concerns


CRM
Answer customer enquiries.
Call center
Complaint
Search
I want to minimize my environmental impact.
I want to drive from A to B.
Available
Save costs & energy
Car pool
Share
Support
Register
Participate
Advertise
Rent
We want to contribute to sustainable mobility.
Car Sharing Community
WebApp
System scope
Data Base
Sustainability?
Profit?
Management
Reporting
Rental
Billing / Statistics
Fill up gas, clean, repair
Return
Parking
High availability
Easy maintenance
Administration
Service team/ Fleet management
Business context
Operational context

Usage Model

Usage Model

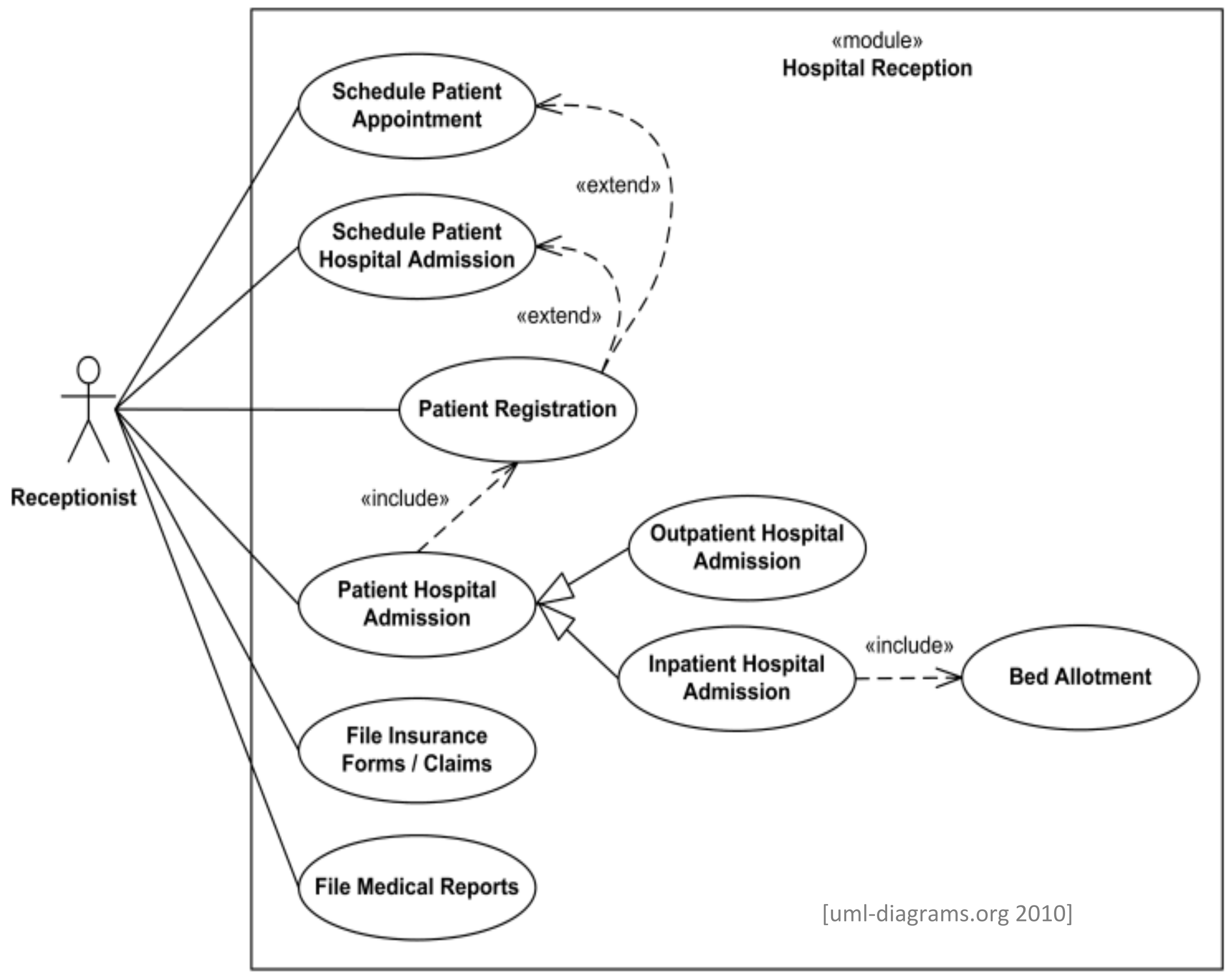
- Def.: A usage model describes the system behavior form the point of view of the user („Black box“) by modeling interaction sequences.
- It specifies the use cases (from the system vision)
- Why? Understanding of intended uses the system.
- Notations:
  - Use case overview diagram
  - Structured text (templates)
  - UML activity diagrams
  - Message Sequence Charts


Context / Environment / Problem Domain
Stakeholder Model
find sustainability stakeholders
elicit sustainability objectives
Objectives, & Goals
Constraints & Rules
Domain Model
analyse sustainability of context
elicit sustainability constraints
Requirements / System / Solution Domain
derive sustainable system vision
System Vision
specify sustainable interaction
Usage Model
Deployment Requirements
System Constraints
Quality Requirements
Process Requirements
refine and deduce sustainability requirements

Use cases & Scenarios

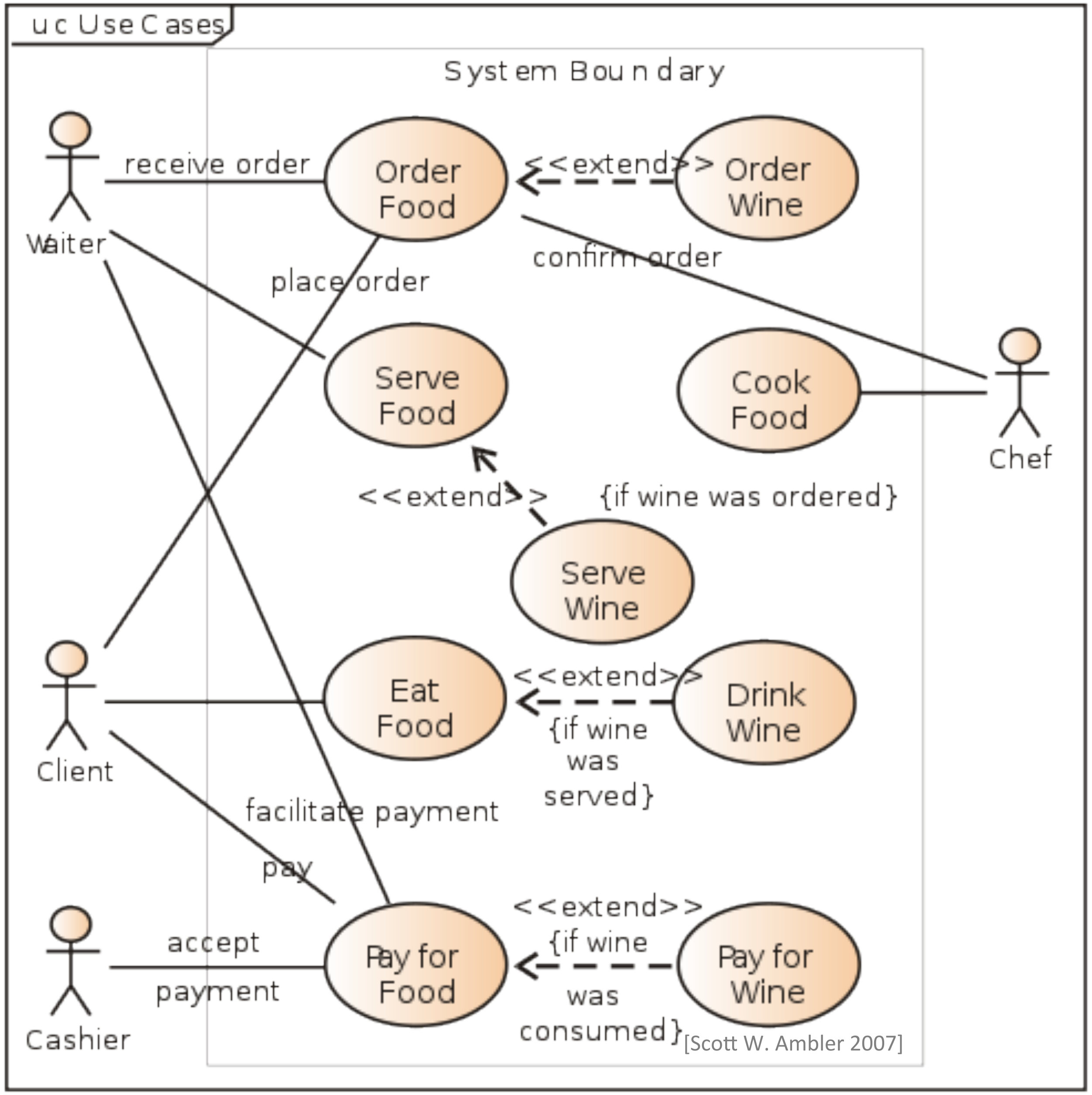
- Def.: A use case is a series of system events triggered by an actor that leads to results for the actor.
- Def.: A scenario is an ordered set of interactions between partners, usually a system and a group of external actors.
- A Usage Model in AMDiRE has three parts:
  - Use Case Overview Diagram („bubble“ diagram)
  - Use Case Templates (one per „bubble“)
  - Scenario diagrams (one per use case template)

Use Case Overview Diagram


«module»
Hospital Reception
Schedule Patient Appointment
Schedule Patient Hospital Admission
«extend»
«extend»
Patient Registration
Receptionist
«include»
Patient Hospital Admission
Outpatient Hospital Admission
Inpatient Hospital Admission
«include»
Bed Allotment
File Insurance Forms / Claims
File Medical Reports

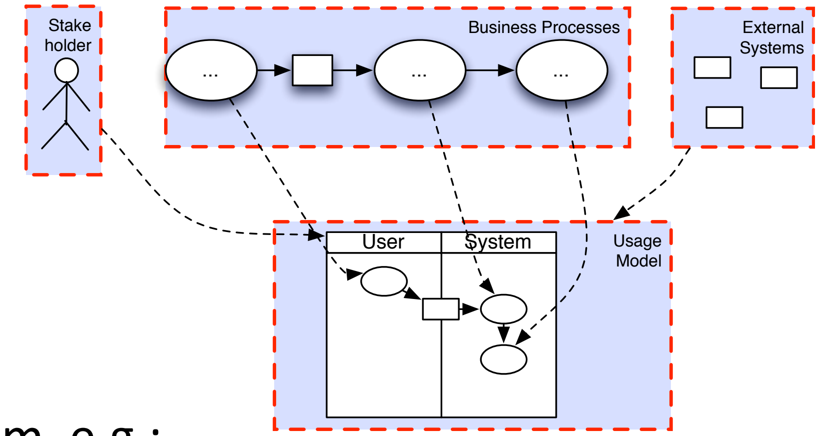
[uml-diagrams.org 2010]

Another Use Case Overview Diagram

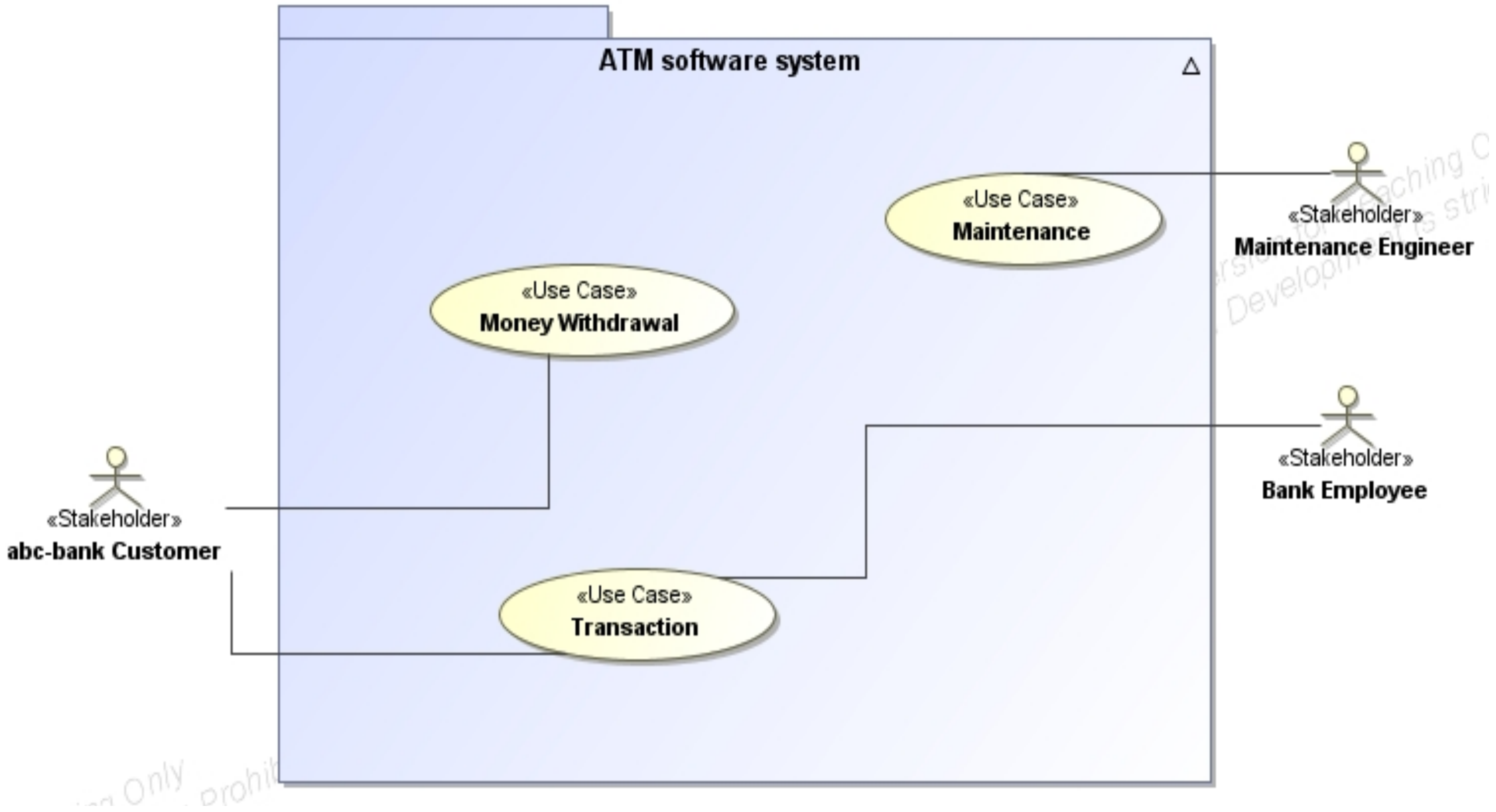


Elaboration of a Usage Model

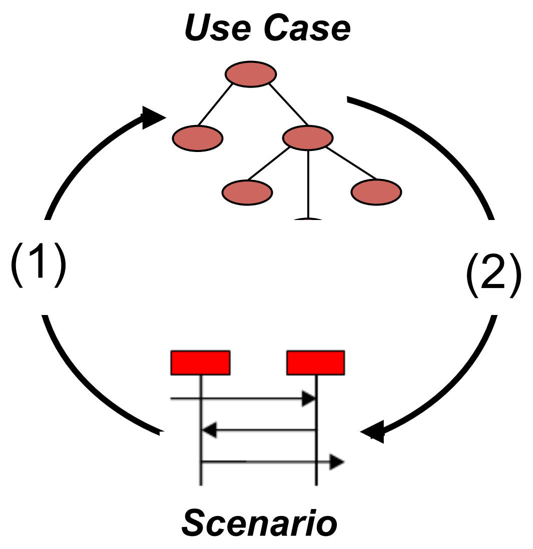
- Deducing the Use Cases
  - Identification of Use Case by business processes
  - Analysis of business processes
    - Task distribution to actors
    - Identification of usage functions
    - Definition of the role of the system, e.g.:
      - Passive support (data administration),
      - Active support (task performance)
  - Informal start: What are the system features?
- Stepwise description and refinement of the scenarios and their interaction
  - Focus on analysis and modeling of
    - Information flow (for later data modeling)
    - Interaction and control flow at the system border



ATM Use Case Overview



Relation: Use Cases and Scenarios



- For each „bubble“ in the overview diagram:
- Use Cases summarize a set of scenarios to a specific usage of the system.
  - Use Case:
    Task, objective, causal relation (pre- and post-conditions)
  - Scenario:
    Sequence of Events (steps, events, interaction)

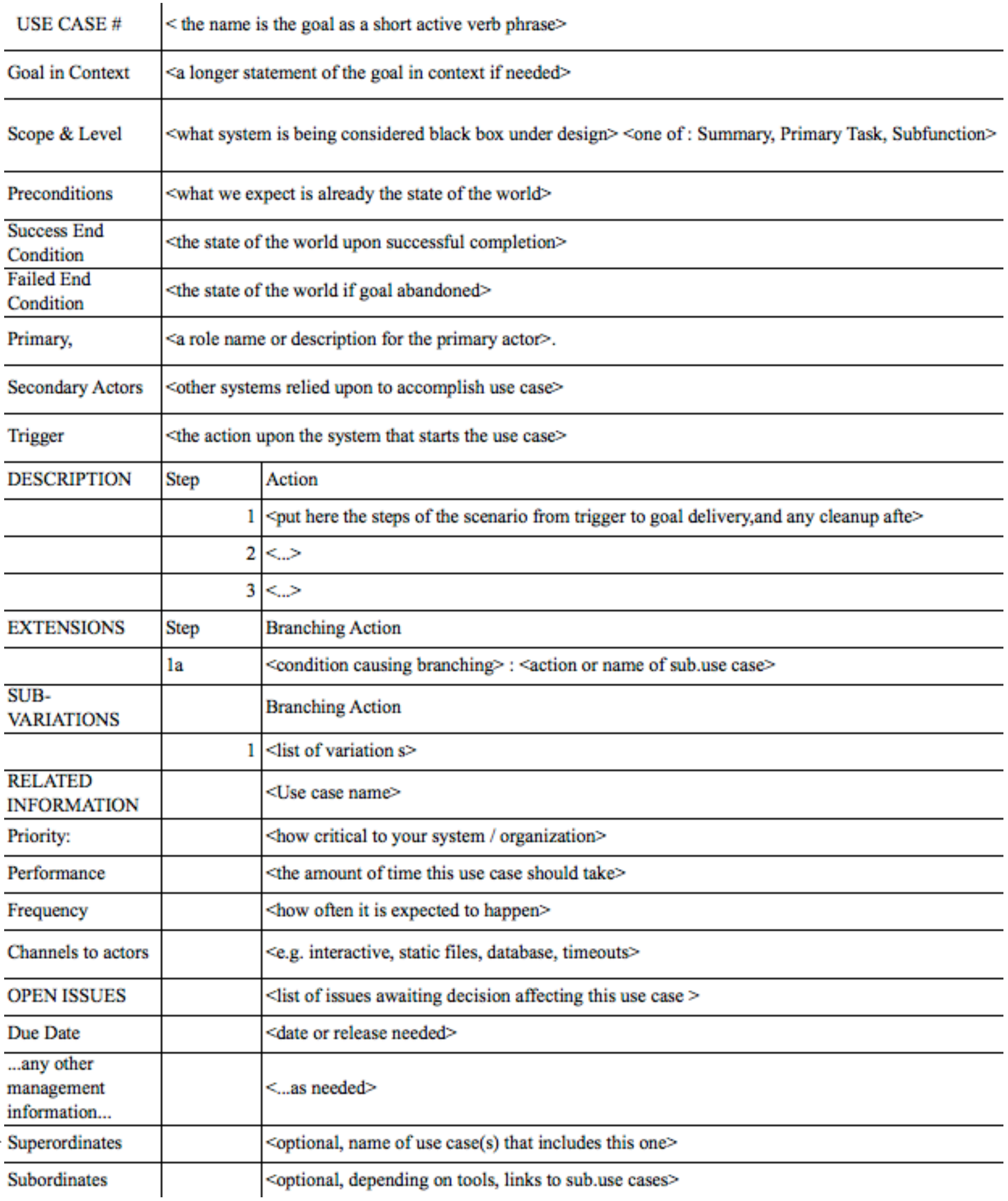
Iterative Elaboration
(compare to refinement and abstraction of goals in the earlier lecture)

(1) Cluster scenarios to tasks

(2) Elicit task-specific scenarios, analyse and walk through them

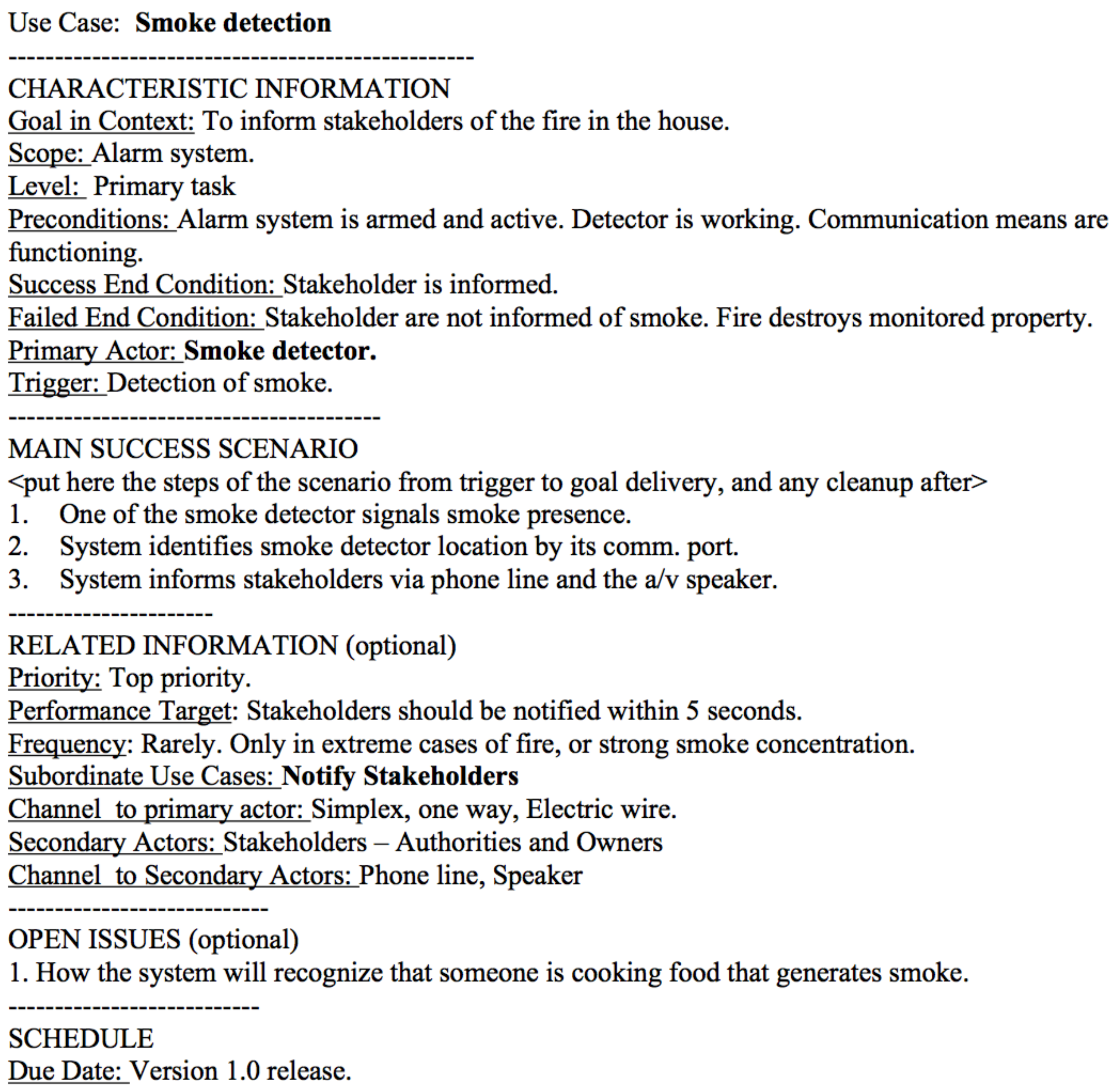
Use cases & Scenarios: Cockburn template

- **Use**: Use cases and scenarios complement each other.
- **Techniques**: Structured text and/or sequence/ interaction diagrams
- **Elicitation**: iterative; combine scenarios to tasks, „play out" task-specific scenarios and analyse



USE CASE #	< the name is the goal as a short active verb phrase>	
Goal in Context	<a longer statement of the goal in context if needed>	
Scope & Level	<what system is being considered black box under design> <one of : Summary, Primary Task, Subfunction>	
Preconditions	<what we expect is already the state of the world>	
Success End Condition	<the state of the world upon successful completion>	
Failed End Condition	<the state of the world if goal abandoned>	
Primary,	<a role name or description for the primary actor>.	
Secondary Actors	<other systems relied upon to accomplish use case>	
Trigger	<the action upon the system that starts the use case>	
DESCRIPTION	Step	Action
	1	<put here the steps of the scenario from trigger to goal delivery,and any cleanup afte>
	2	<...>
	3	<...>
EXTENSIONS	Step	Branching Action
	1a	<condition causing branching> : <action or name of sub.use case>
SUB-VARIATIONS		Branching Action
	1	<list of variation s>
RELATED INFORMATION		<Use case name>
Priority:		<how critical to your system / organization>
Performance		<the amount of time this use case should take>
Frequency		<how often it is expected to happen>
Channels to actors		<e.g. interactive, static files, database, timeouts>
OPEN ISSUES		<list of issues awaiting decision affecting this use case >
Due Date		<date or release needed>
...any other management information...		<...as needed>
Superordinates		<optional, name of use case(s) that includes this one>
Subordinates		<optional, depending on tools, links to sub.use cases>

Example



Use Case: **Smoke detection**

CHARACTERISTIC INFORMATION

Goal in Context: To inform stakeholders of the fire in the house.
Scope: Alarm system.
Level: Primary task
Preconditions: Alarm system is armed and active. Detector is working. Communication means are functioning.
Success End Condition: Stakeholder is informed.
Failed End Condition: Stakeholder are not informed of smoke. Fire destroys monitored property.
Primary Actor: **Smoke detector.**
Trigger: Detection of smoke.

MAIN SUCCESS SCENARIO

<put here the steps of the scenario from trigger to goal delivery, and any cleanup after>

1. One of the smoke detector signals smoke presence.
2. System identifies smoke detector location by its comm. port.
3. System informs stakeholders via phone line and the a/v speaker.

RELATED INFORMATION (optional)

Priority: Top priority.
Performance Target: Stakeholders should be notified within 5 seconds.
Frequency: Rarely. Only in extreme cases of fire, or strong smoke concentration.
Subordinate Use Cases: **Notify Stakeholders**
Channel to primary actor: Simplex, one way, Electric wire.
Secondary Actors: Stakeholders – Authorities and Owners
Channel to Secondary Actors: Phone line, Speaker

OPEN ISSUES (optional)

1. How the system will recognize that someone is cooking food that generates smoke.

SCHEDULE

Due Date: Version 1.0 release.

1 of 2

- **Use Case:** <number> <the name should be the goal as a short active verb phrase>
- CHARACTERISTIC INFORMATION
  - Goal in Context: <a longer statement of the goal, if needed>
  - Scope: <what system is being considered black-box under design>
  - Level: <one of: Summary, Primary task, Subfunction>
  - Preconditions: <what we expect is already the state of the world>
  - Success End Condition: <the state of the world upon successful completion>
  - Failed End Condition: <the state of the world if goal abandoned>
  - Primary Actor: <a role name for the primary actor, or description>
  - Trigger: <the action upon the system that starts the use case, may be time event>
- MAIN SUCCESS SCENARIO
  - <put here the steps of the scenario from trigger to goal delivery, and any cleanup after>
  - <step #> <action description>
- EXTENSIONS
  - <put here there extensions, one at a time, each refering to the step of the main scenario>
  - <step altered> <condition> : <action or sub.use case>
  - <step altered> <condition> : <action or sub.use case>
- SUB-VARIATIONS
  - <put here the sub-variations that will cause eventual bifurcation in the scenario>
  - <step or variation # > <list of sub-variations>
  - <step or variation # > <list of sub-variations>

2 of 2

- RELATED INFORMATION (optional)
  - Priority: <how critical to your system / organization>
  - Performance Target:
    <the amount of time this use case should take>
  - Frequency: <how often it is expected to happen>
  - Superordinate Use Case:
    <optional, name of use case that includes this one>
  - Subordinate Use Cases:
    <optional, depending on tools, links to sub use cases>
  - Channel to primary actor:
    <e.g. interactive, static files, database>
  - Secondary Actors:
    <list of other systems needed to accomplish use case>
  - Channel to Secondary Actors:
    <e.g. interactive, static, file, database, timeout>
- OPEN ISSUES (optional)
  - <list of issues about this use cases awaiting decisions>
- SCHEDULE
  - Due Date: <date or release of deployment>

Example Use Case ATM

- **Use Case:** 1 withdraw money
- CHARACTERISTIC INFORMATION
  - Goal in Context: user withdraws money from the ATM
  - Scope: ATM
  - Level: Primary task
  - Preconditions: user has an ATM card and has access to ATM
  - Success End Condition: user gets money
  - Failed End Condition: user doesn't get money
  - Primary Actor: customer (= user)
  - Trigger: ATM card entered by user
- MAIN SUCCESS SCENARIO
  1. User enters card
  2. System prompts for PIN
  3. User enters PIN
  4. System prompts options for withdrawal / transfer / deposit money
  5. User selects withdraw
  6. System prompts for amount
  7. User enters amount
  8. System returns money
- EXTENSIONS
  - 5. *condition* selection of different account: *action* Withdraw from different account
  - <step altered> <condition> : <action or sub.use case>
  - <step altered> <condition> : <action or sub.use case>
- SUB-VARIATIONS
  - 4. *condition* user entered wrong PIN: *action* system displays error message
  - 8. not enough money: system displays error message
  - <step or variation # > <list of sub-variations>
- RELATED INFORMATION (optional)
  - Priority: critical
  - Performance Target: one minute
  - Frequency: very often (depends on location of ATM)
  - Superordinate Use Case: <optional, name of use case that includes this one>
  - Subordinate Use Cases: <optional, depending on tools, links to sub.use cases>
  - Channel to primary actor: interactive
  - Secondary Actors: <list of other systems needed to accomplish use case>
  - Channel to Secondary Actors: <e.g. interactive, static, file, database, timeout>
- OPEN ISSUES (optional)
  - <list of issues about this use cases awaiting decisions>
- SCHEDULE
  - Due Date: May 2014

Todos

System Vision

Usage Model

Submit both to me as one PDF file per team by the end of today.

Birgit Penzenstadler
birgit.penzenstadler@csulb.edu
www.csulb.edu/~bpenzens
@twinkleflip
#SustainabilityDesign #KarlskronaManifesto