

Requirements Engineering: Goals and Constraints

CECS 590

Visit by Intel

- March 18 or 23 (to be confirmed)
- Erik Simmons, head of “Emergent Systems and Coaching”
- What would you like him to talk about?
(idea: 30 min presentation plus long Q & A)
- I will give you his slide set a week before so you can prepare for the visit
→ flipped classroom

Literature Review

- Feedback posted on BeachBoard
- Points assigned
- Remember: You all made an effort, and I recognize that. I wrote more about my critique and things you could do better than about the things that worked well – because that's what you need to know to improve.

References – again...

In one of the reference lists:
“XML messaging for Mobile Devices: From requirements to implementation”
by Jakko Kangasharju, Sasu Tarkoma
published in *ScienceDirect*

[\[HTML\]](#) [XML messaging for mobile devices: From requirements to implementation](#)

[sciencedirect.com](#) [\[HTML\]](#)

[J Kangasharju](#), [T Lindholm](#), [S Tarkoma](#) - [Computer Networks](#), 2007 - Elsevier

Get it! [@CSULB](#)

In recent years, both the number and capabilities of **mobile devices** have increased rapidly to the point where the **mobile** world is becoming a significant part of the Internet. Another recent trend is the increase in **XML** use for communication between applications. However ...

[Cited by 21](#) [Related articles](#) [All 3 versions](#) [Web of Science: 6](#) [Cite](#) [Save](#)

References – again...

Cite

Copy and paste a formatted citation or use one of the links to import into a bibliography manager.

MLA Kangasharju, Jaakko, Tancred Lindholm, and Sasu Tarkoma. "XML messaging for mobile devices: From requirements to implementation." *Computer Networks* 51.16 (2007): 4634-4654.

APA Kangasharju, J., Lindholm, T., & Tarkoma, S. (2007). XML messaging for mobile devices: From requirements to implementation. *Computer Networks*, 51(16), 4634-4654.

Chicago Kangasharju, Jaakko, Tancred Lindholm, and Sasu Tarkoma. "XML messaging for mobile devices: From requirements to implementation." *Computer Networks* 51, no. 16 (2007): 4634-4654.

[BibTeX](#) [EndNote](#) [RefMan](#) [RefWorks](#)

Your team projects

- Software Scalable Three-Dimensional Printer Carriage w/ Interchangeable Print Heads
- Home automation installation work order system
- Pitching machine for tennis balls
- PriceShare
- HyperSocialYelp
- StuList
- Product Review on Social Platform
- Sportsbuddy
- Beach Cops
- 2 tbd

Recap time!

- Stakeholders
 - Definition
 - What types of stakeholders are there?
What roles? Which functions?
 - Which are the characteristics we need for stakeholder analysis?
 - Why a stakeholder model?
How do I elaborate a stakeholder model?

Requirements Engineering – Outline

- WHY do we need Requirements Engineering and what is it?
- Principles: Definitions, process, roles, problem/solution view, artifact orientation
- System Models: Decomposition and abstraction, system views
- Frameworks: What reference structures can I use for requirements?
- Business Case Analysis: Why are we building this system?
- Stakeholders: Who are the people to talk to about requirements?
- **Goals and Constraints: What are the major objectives for the system?**
- System Vision: What exactly do we want to achieve?
- Domain Models: What are the surrounding systems ours interacts with?
- Usage Models: How will the system interact with the user?
- Software quality models: How to determine the quality characteristics?
- Quality requirements: How to specify which qualities need to be met?
- Process requirements: How to specify constraints for development?
- Towards a system specification: How to hand over to design?
- Quality assurance: How to ensure that RE is done in a good way?
- Change management: How to evolve requirements?

Connecting to last lecture: Stakeholders and Elicitation



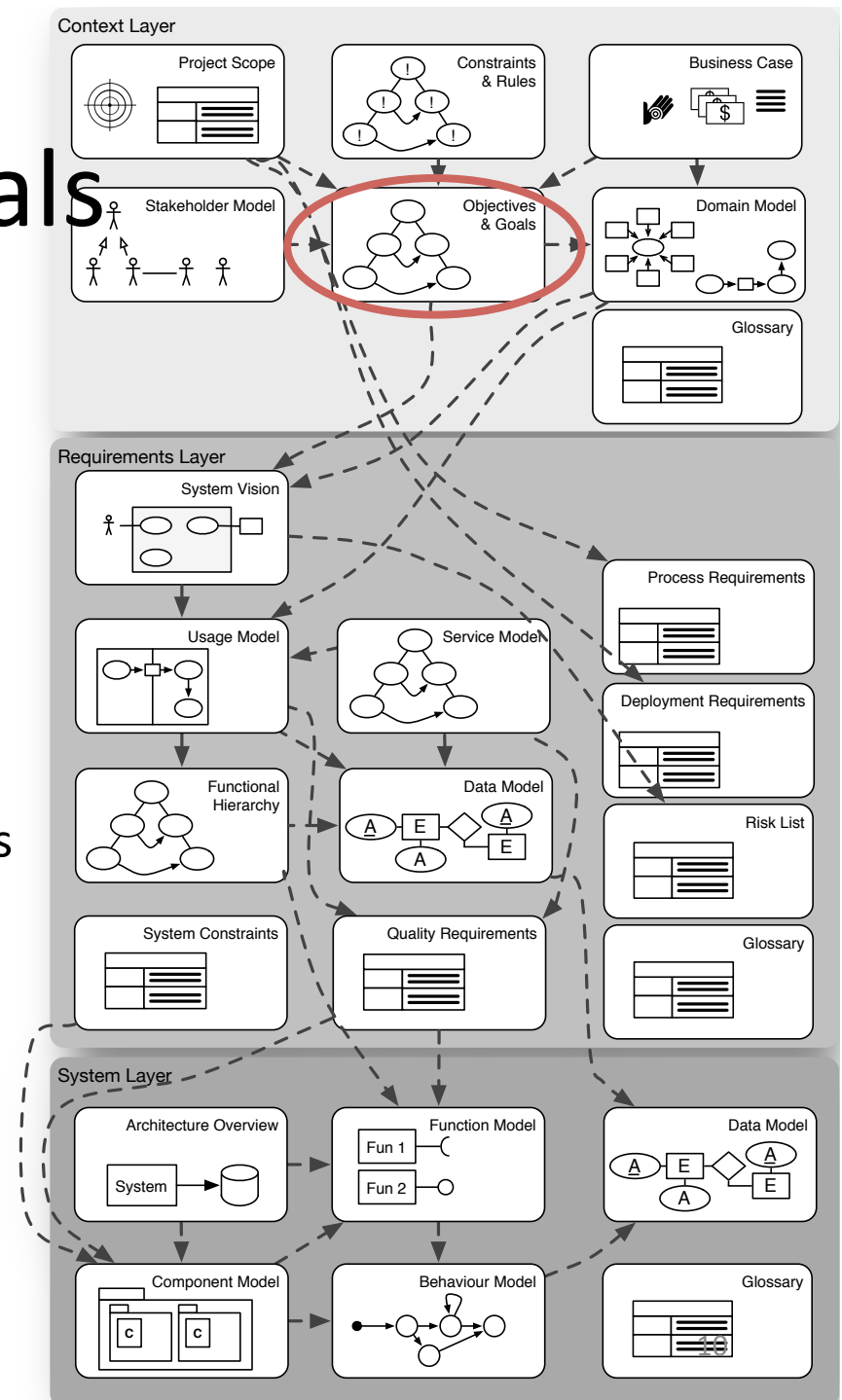
- Elaboration of an adequate, complete requirements and system specification
 - Stakeholder Model
 - Elicitation and creativity techniques
 - Modeling is part of requirements elicitation and analysis

Questions that follow w.r.t. goals:

- How to elicit goals in models and which techniques are available?
- Which interdependencies exist between goal models and other RE contents?
- How can I use goal models within the requirements analysis?
- Which challenges may arise when performing goal modeling?

Today's learning goals

- **Foundations of goal modeling**
 - Definition, characteristics, meaning and purpose
 - Interdependencies between goal models and other RE content
- **Overview of**
 - Types of goals
 - Goal models in the context of analysis (dependencies, identification and avoiding conflicts, QA)
 - Techniques for goal modeling
- Goal modeling in context of **AMDiRE**
- **Understanding** of typical problems



Goals and Constraints

- Motivation and overview
- Definition
- Types of goals
- Goal refinement and abstraction
- Goal models and usage for analysis and quality assurance
- Techniques for goal modeling
- Goal modeling with AMDiRE
- Typical problems when using goal models

Meaning of explicit goals

- Explicit elicitation of the goals pursued by a **project**/ a system development/ a **system** :

Why should the system be developed?

→ Which **goals** are being pursued?

- Business goals, market goals, relation to strategic goals of the company, ...
- Technical goals
- Methodical goals
- Functional goals
- ...

→ Goals form an **essential part of the rationale for requirements** and build a basis early on for

- **Traceable** requirements
- **Prioritization** of requirements
-

Problems addressed by goal modeling

Advantages of a systematic elicitation and documentation of explicit, negotiable, assessable goals:

- **Analysis** of goals as well as
 - early **identification of goal conflicts**
 - early **identification of „Moving Targets“**
 - A series of requirements **justifications** documents the decision path from goals to solution basis.
 - **Differentiation between problem and solution:**
 - How are the requirements to be evaluated and to be justified?
 - What are the necessary and reasonable assumptions?
 - What reveals itself as free (unconstrained) design decision?
- **Avoid** too much solution orientation (**via goal abstraction**).

Examples for goals

- Goal of the system user or customer:
 - *The system administrates the essential data of our company over the next 20 years.*
- Business goal of the system manufacturer or an investor:
 - *The system reaches a market satisfaction of at least 20%.*
- Quality goals of a typical user:
 - *The system is easy to handle.*
 - *The system is fast and efficient.*
- Prescriptions by a standard:
 - *The system fullfills the security standard xyz.*

Which goals could the stakeholders have?



Goals and Constraints

- Motivation and overview
- Definition
- Types of goals
- Goal refinement and abstraction
- Goal models and usage for analysis and quality assurance
- Techniques for goal modeling
- Goal modeling with AMDiRE
- Typical problems when using goal models

What is a goal?

Definition

Discretionary abstract characteristic, which

1. the system shall fulfill w.r.t. it's operational environment
or
2. the development process of the system shall fulfill.

Critical points

- No hint w.r.t. the solution and often not quantified.
 - Unmistakeable clarity, when a goal is reached
 - Basic achievability of a goal

Delimitation of goals and requirements

„A goal is a prescriptive statement of intent.“

– Axel van Lamsweerde

→ Goals impose **general constraints** or a vision, state or effect that is connected with development.

By the way:

- „Prescriptive“: A (high-level) goal itself often doesn't require a rationale, goals are the rationale for deduced requirements.
- Goals serve for the understanding of the to-be-reached state of the system, the system context, etc.

Example

- Goals are used for gap analysis and conversion from the current state of business processes and legacy systems to a desired state.

From goals to requirements

It is hard to define a clear delimitation between goals and requirements.

In tendency

- **Goals:** encompassing, general, bound to intents and motivations, mostly directed towards the context of the system under development
- **Requirements:** Demands for concrete, often detailed characteristics of a system or it's development process (ideally rationalized by a goal)

Delimitation

- In some approaches (e.g. KAOS) requirements are denoted as specialized goals that are related to specific subsystems and actors („Agents“).

Characteristics of goals

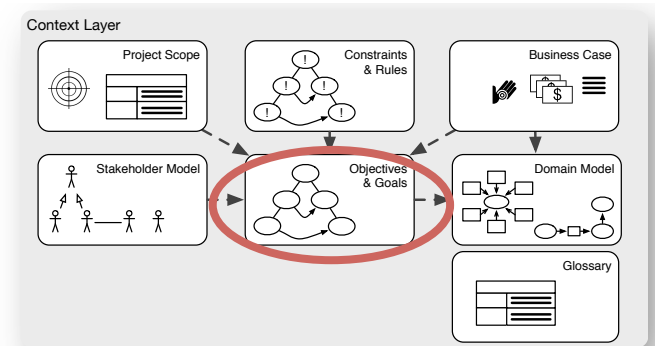
- A goal is characterized by it's **goal description**
- We distinguish between goals and their description
 - **Precision**: How unambiguous is a goal described?
(subjective/objective goal description)
 - **Concreteness/abstractness**
 - **Verifiability**: How unambiguously can the satisfaction of the goal be verified?
 - **Degree of satisfaction**: Is the goal satisfiable to a specific degree or only as a whole?

Careful: Often goals are characterized indirectly by metrics that shall be achieved!

Goal satisfaction

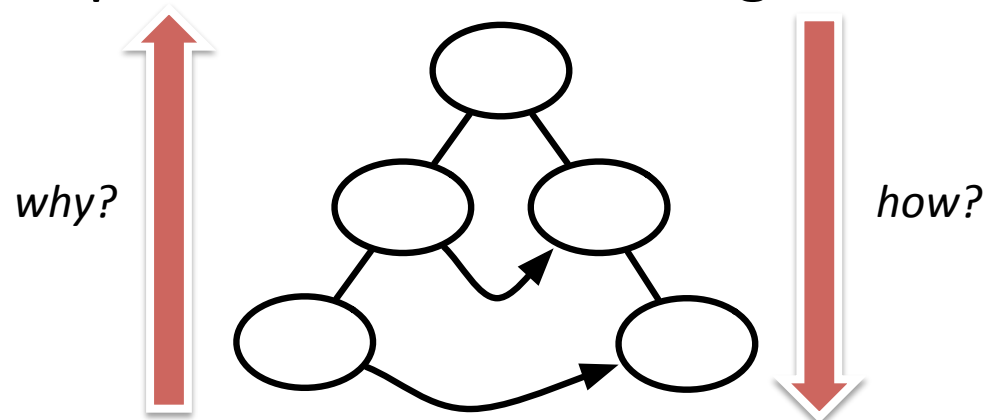
- **Satisfaction of goals** happens via adequate steps, characterized by
 - subgoals
 - (deduced) requirements / constraints
- To **characterize** goals and make them more precise, it is helpful to introduce metrics for the degree of satisfaction
 - Relations between goals, subgoals via formulas
 - This allows for statements whether satisfying *Goal A* to a higher degree helps, hinders or doesn't influence *Goal B*
 - Definition of performance metrics for individual goals (for example via GQM – the Goal Question Metric approach)

Goal models: Idea

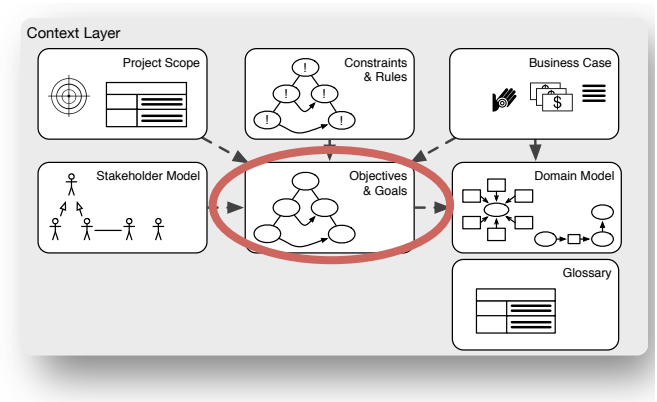


Goal models for structuring all relevant:

- Goals and subgoals
- Relations between goals:
 - Hierarchy/decomposition of goals and/into subgoals
 - Interdependencies between goals



Goal models



- **Network of characteristics**
 - Set of related characteristics
- **Goal tree**
 - Set of related characteristics of goals with a hierarchical refinement relation and different interdependencies
 - Known representatives: KAOS (van Lamsweerde, et al.), GSN (Goal Structuring Notation, Kelly et al.), i* (Yu et al.), ...
- **Expectation**
 - Leaf of the goal tree that can exclusively be related to it's context with regard to it's satisfaction.
- **Requirement**
 - Leaf of the goal tree that can exclusively be related to the system with regard to it's satisfaction.
 - Functional or non-functional characteristic of the system that can be deduced from the goals
- Expectation and requirement can be combined to an **Assumption/Guarantee pair** (A/G specification) w.r.t. the interface between system and context.

Goals and Constraints

- Motivation and overview
- Definition
- Types of goals
- Goal refinement and abstraction
- Goal models and usage for analysis and quality assurance
- Techniques for goal modeling
- Goal modeling with AMDiRE
- Typical problems when using goal models

Types of goals

- **Categorization** of goals happens according to different **criteria**:
 - Underlying **system models**, **quality models**, ...
 - **Application domain/** Characteristics of the expected **system context**
 - **Methodical approach** (e.g. KAOS) as well as **characteristics** that can be expressed by modeling concepts



Examples for types of goals according to Lamsweerde

- Quality-related goals (part of non-functional goals)
 - Usability, Performance, Security, Accuracy
 - *Reliability: 99% of the uptime, the system reacts correctly to user queries.*
- Optimization goals (also „soft“ goals)
 - If measurable: Max-/Minimization, increase/decrease
 - *Performance: The operating grade of the tracks between A and B is increased by 3% per year in the next 5 years.*
- Behavioral goals (also „hard“ or functional goals):
 - E.g. after interaction phase: satisfaction goal (A requests from B), information goal (A informs B)
 - *Accuracy: 280-300ms, the train control must be informed about the state of the level-crossing.*
- Anti-goals (also „obstacles“):
 - E.g. „safety hazards“ and „security threats“
 - *The data loss is estimated at 5% despite backup.*

Exercise

For each of the following types, specify two goals for a car sharing system:

- Quality-related goals (part of non-functional goals)
- Optimization goals (also „soft“ goals)
- Behavioral goals (also „hard“ or functional goals)
- Anti-goals (also „obstacles“)

Goals and Constraints

- Motivation and overview
- Definition
- Types of goals
- Goal refinement and abstraction
- Goal models and usage for analysis and quality assurance
- Techniques for goal modeling
- Goal modeling with AMDiRE
- Typical problems when using goal models

Ideal RE: Concretion and Abstraction

Ideally, **RE Top Down** happens in this form:

- Based on elicited and consolidated, generally formulated goals
 - *The product must be deployable internationally*
- We define abstract requirements, consolidated and documented:
 - *The product must be deployable in 3rd world countries*
 - *The product must be usable independent of the energy resource*
- And retrieve refined, concrete requirements
 - *The system has a dynamo, that generates the necessary energy; the electric tension is at...*
- And finally determine a system specification
 - *We use a dynamo of the company AB...*
- Reverse path: From system specification /concrete requirements or prototypes to goals (Reverse Engineering)

Example (simplified)

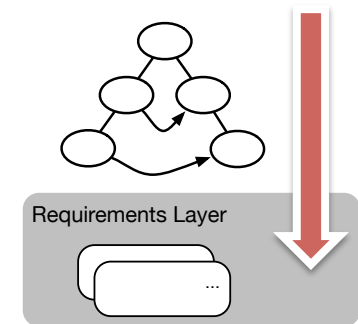
Goals: (Often) hard to measure

1. *The system must be deployable internationally.*
2. *The system must be easy to use.*
3. *The development will follow a systematic process.*

Requirements: objectively measurable/decideable

1. *The user interface must support the languages [x,y,z]..*
2. *The average inexperienced user must be able to use the function [Function] within 20 seconds.*
3. *The process must be certified as CMMI Level 3.*

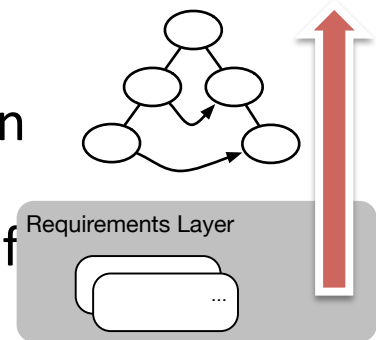
- **Roundtrip:** requirements make the satisfaction of goals more concrete (over various steps), goals describe a justification for requirements.
- **Tracing:** Ideally, the relations between goals, abstract and concrete requirements are documented in a traceable way.



Goal abstraction and goal refinement

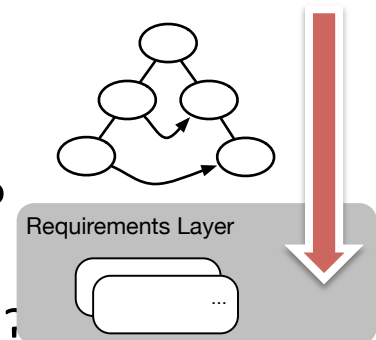
Goal abstraction

- What goals are behind the requirements and might open up the design space?
 - Which goals are determining the currently present set of requirements?
- „Why“ questions: retrieval of application domain knowledge



Goal refinement (Re decisions) and Scoping:

- Problem scoping: for which tasks or functions of the system do we need requirements?
For which context do we need to deduce requirements?
- System scoping: How can the design space of the problem under consideration be technically determined?



Goals and Constraints

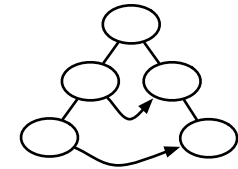
- Motivation and overview
- Definition
- Types of goals
- Goal refinement and abstraction
- Goal models and usage for analysis and quality assurance
- Techniques for goal modeling
- Goal modeling with AMDiRE
- Typical problems when using goal models

Do we have a goal conflict here?



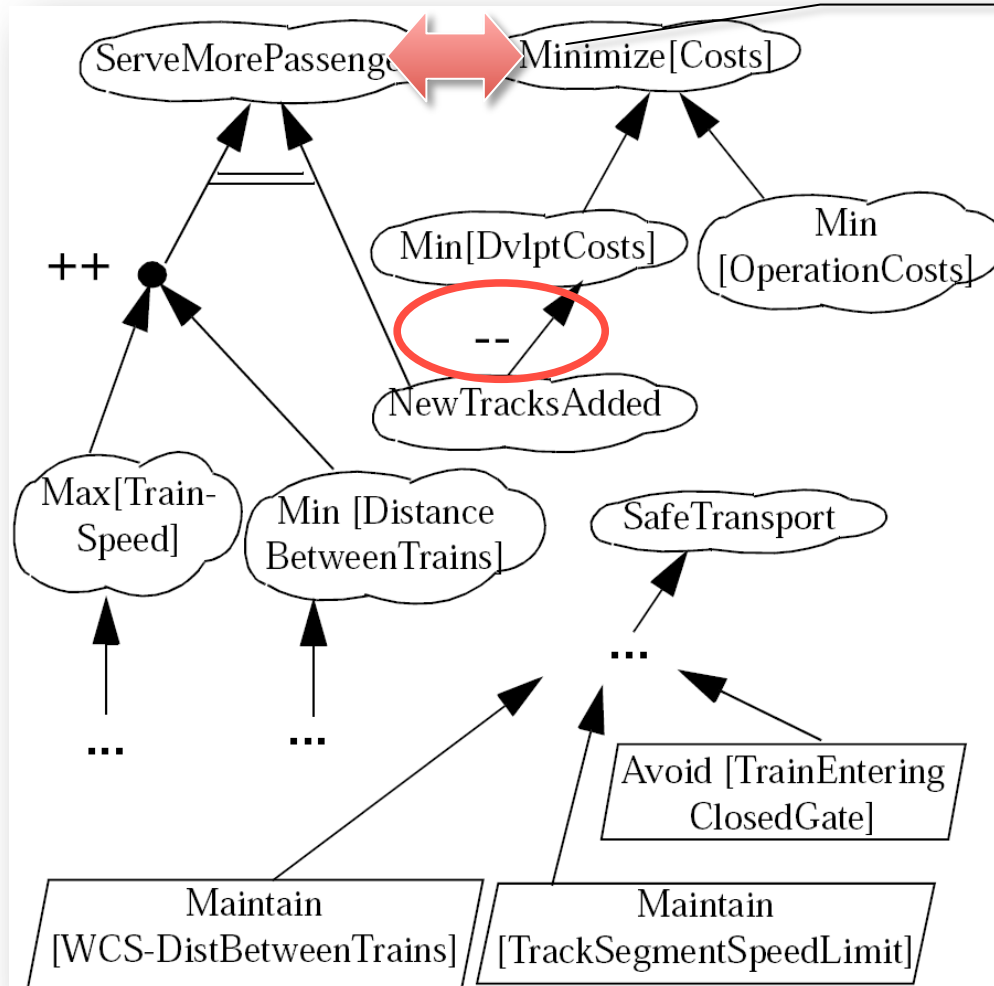
Usage of goal models for conflict analysis

Resolving goal conflicts



1. Goals are made measurable by evaluation and prioritization – and thereby made comparable.
2. For goals, specific threshold levels are determined as standards that can be reached by all.
3. Goals are prioritized.
4. Conflicts are resolved by negotiation.

Identification of goal conflicts in a KAOS example



Conflict between optimization goals

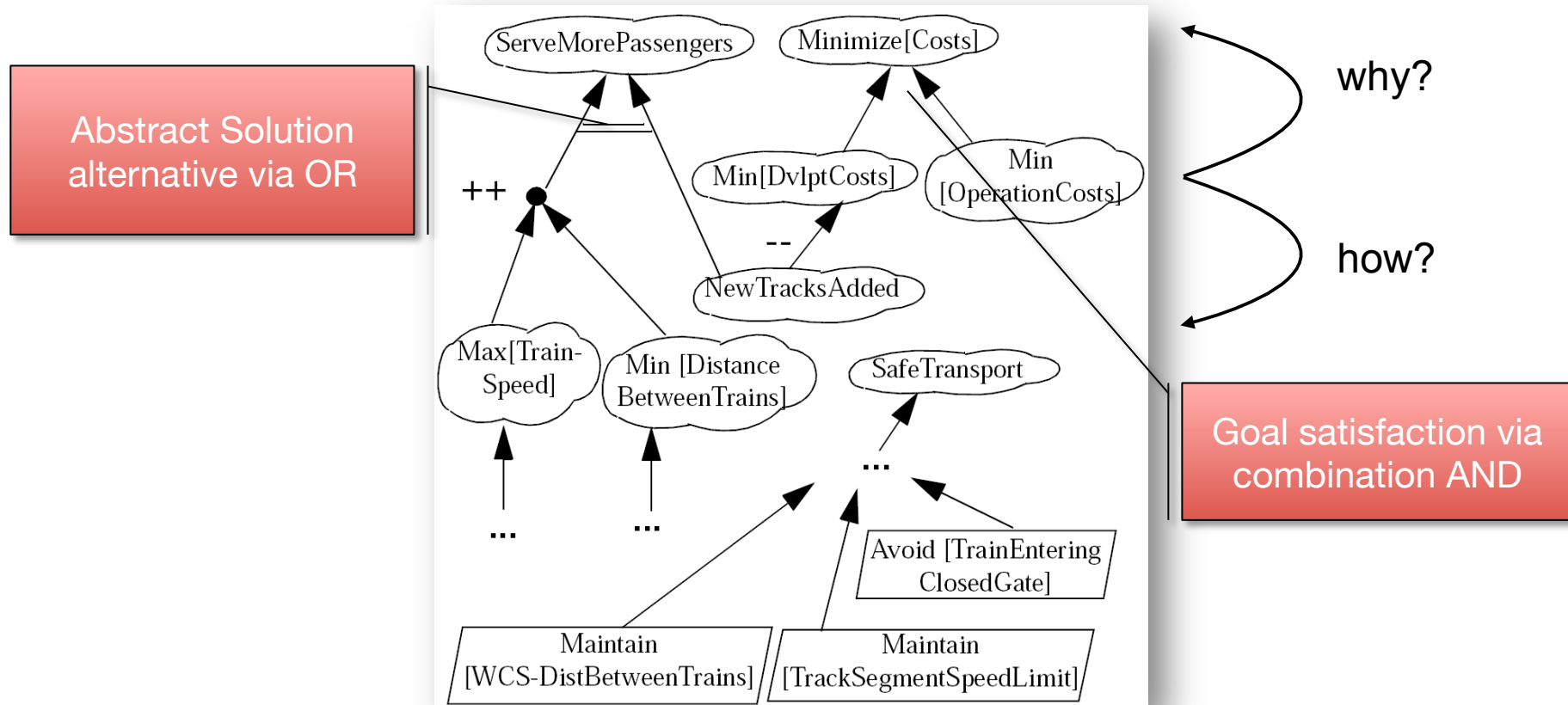
Goals and Constraints

- Motivation and overview
- Definition
- Types of goals
- Goal refinement and abstraction
- Goal models and usage for analysis and quality assurance
- Techniques for goal modeling
- Goal modeling with AMDiRE
- Typical problems when using goal models

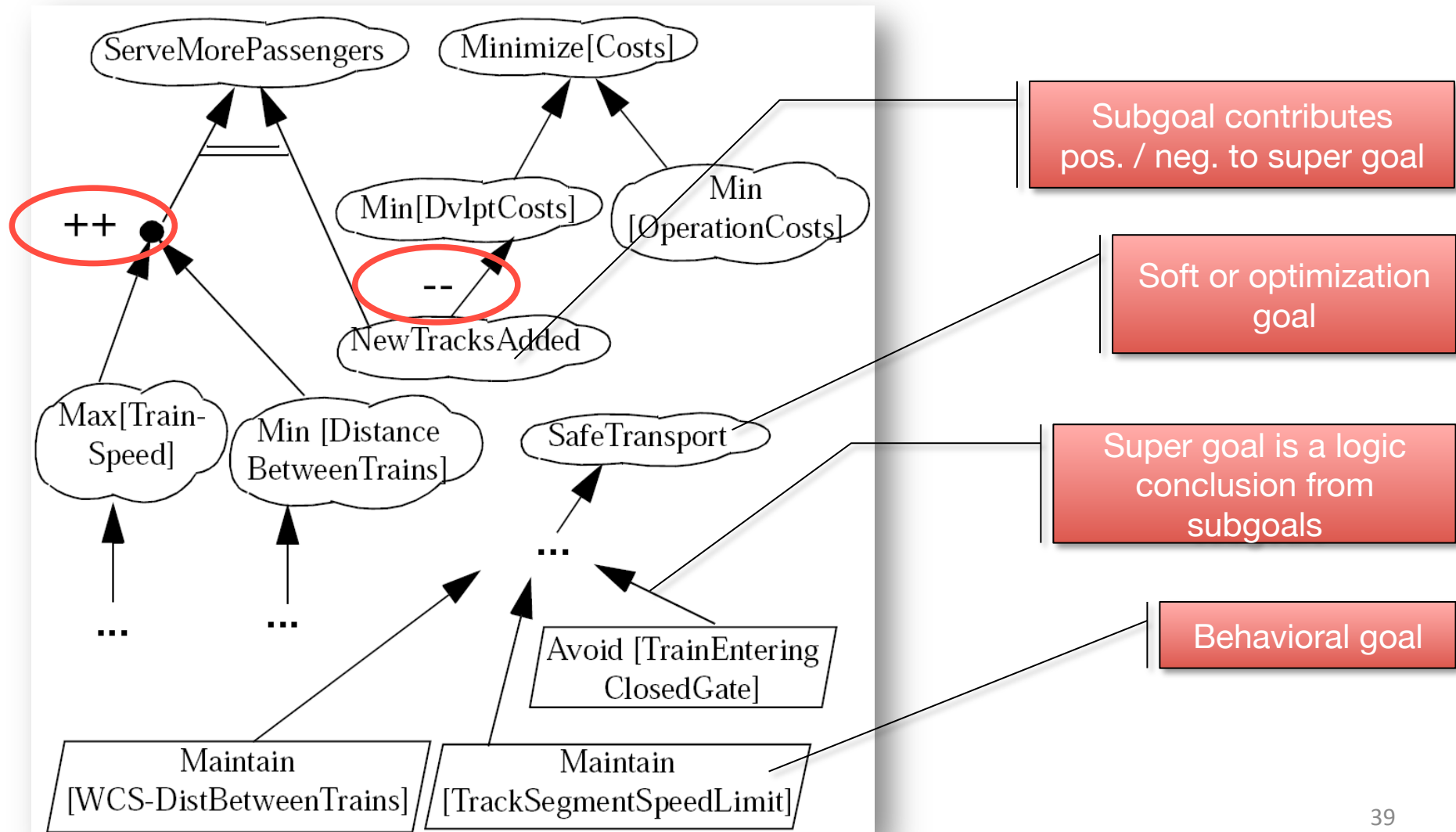
Goal modeling techniques

- KAOS (**K**ee**A**ll **O**bjects **S**atisfied)
- i* Modelling Framework
- *System Dynamics* and *Balanced Scorecards* for encompassing analysis and simulation of complex dynamic systems
- **Basic ideas and commonalities** of goal modeling techniques:
 - Hierarchical decomposition of goals into subgoals
 - Model of dependencies (e.g. conflicts, ...)
 - Mapping of goals to agents (subsystems/components/actors)
 - „seamless transition“ to design

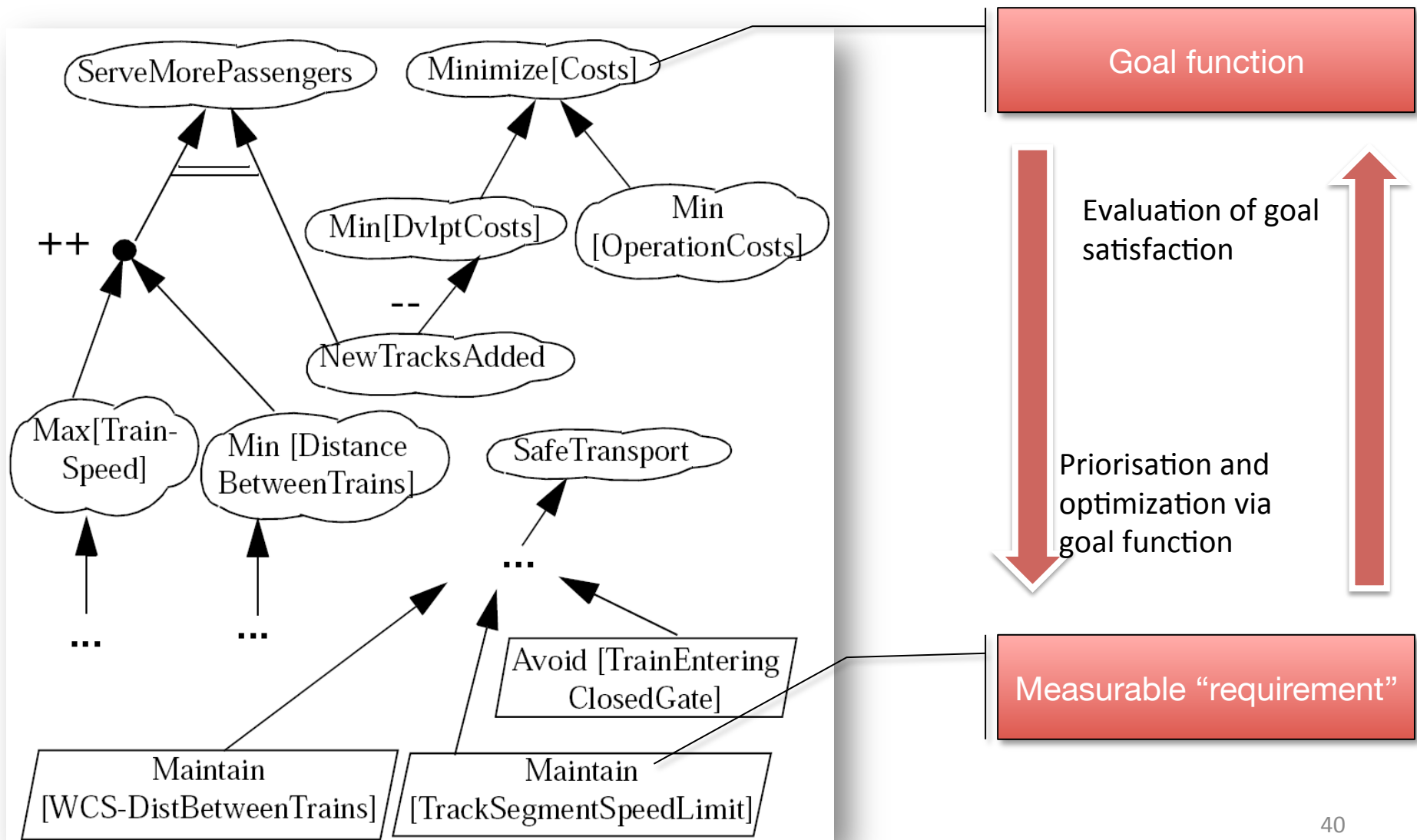
Example technique: KAOS



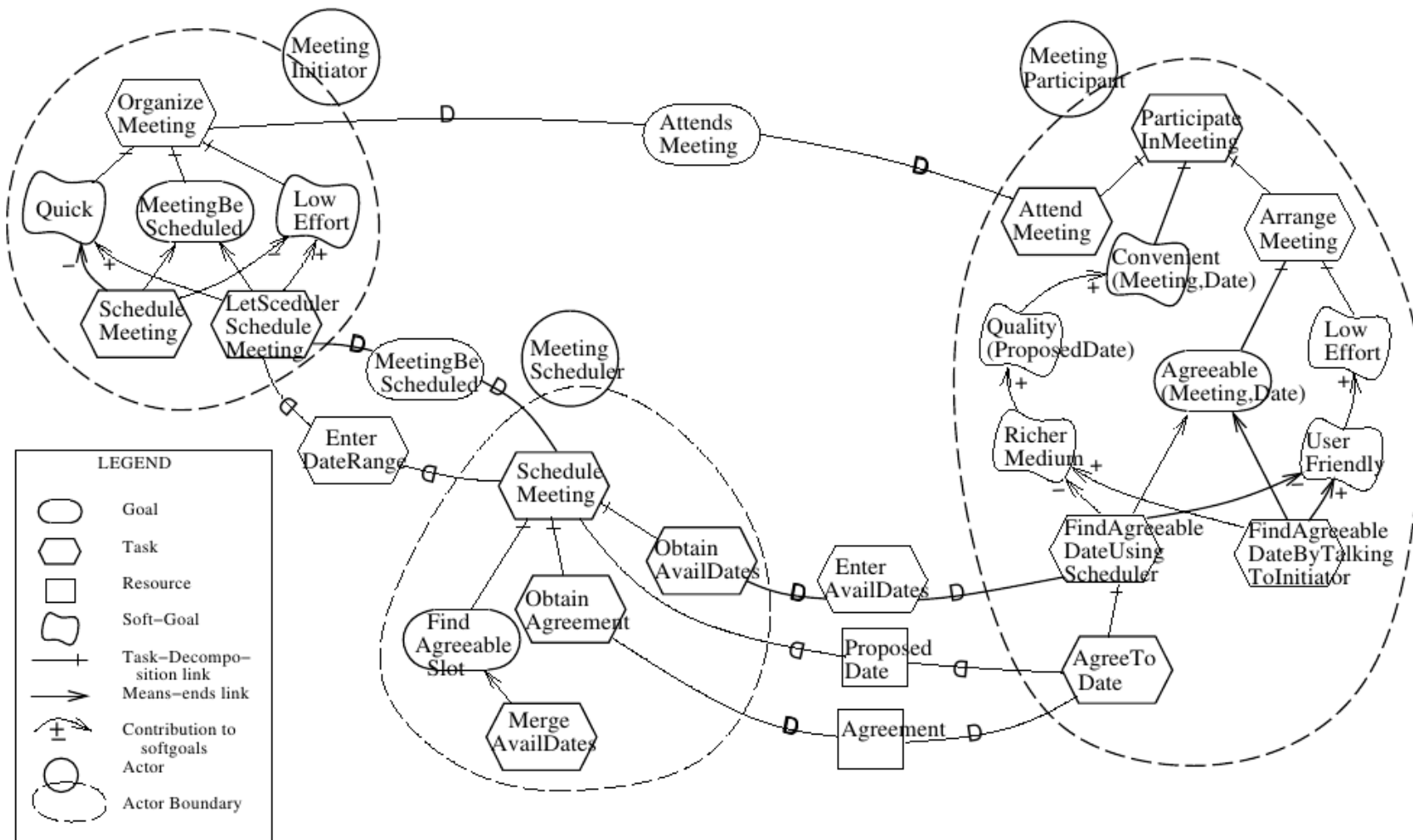
Example technique: KAOS



Measuring goal satisfaction

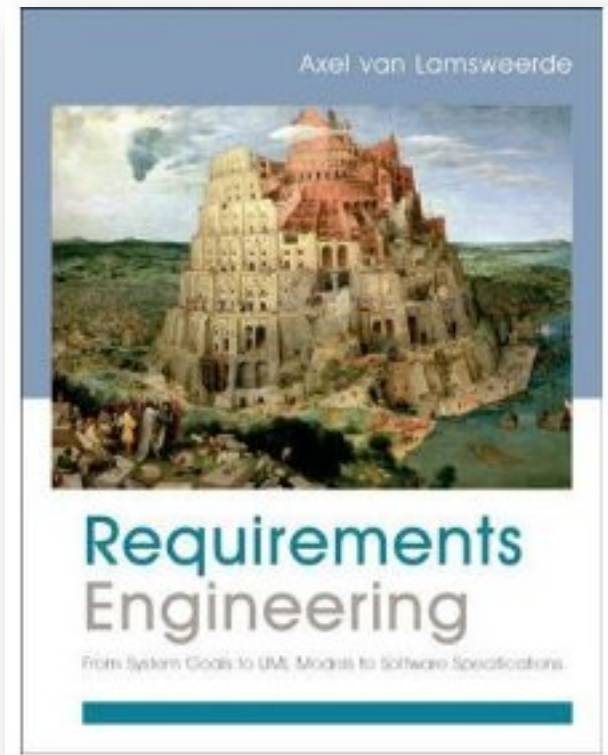


Example technique: i*



Quelle: Yu, siehe ff.

References...



Axel van Lamsweerde: Requirements Engineering -- From System Goals to UML Models to Software Specifications, 2009.

Why Agent-Oriented Requirements Engineering

Eric S. K. Yu

Faculty of Information Studies, University of Toronto
Toronto, Ontario, Canada M5S 3G6
www.cs.utoronto.ca/~eric

Abstract. Agent concepts have been used in a number of recent approaches to requirements engineering. In view of the rapid shift towards open, networked, and cooperative computing, we argue for the fuller development of emerging agent-oriented approaches to requirements engineering. Such approaches would address crucial requirements engineering concerns such as functionality, quality, and process, using *agent* as the focal concept.

1 Introduction

Agent concepts have been used in a number of recent approaches to Requirements Engineering (RE). It is generally acknowledged that the main focus of Requirements Engineering should be on the characterization of the intended system in relation to its environment [2, 12, 14]. Agent concepts have been introduced in RE primarily as modelling constructs to characterize active elements in the environment, usually including the target system. These active elements may be human or machine, and may contain hardware and/or software.

In this paper, we put forth the position that the concept of agent should be further developed to serve as a focal, guiding concept in RE frameworks, much like *objects* and *goals* have served as guiding concepts. The key benefit of having an RE-level concept of agent, and using it as a guiding concept during RE, is that it will serve to bring issues centring on an agent together, so that they can be identified and addressed.

An agent-oriented approach to RE will be of particular interest for new settings in which there is a high degree of open distributed

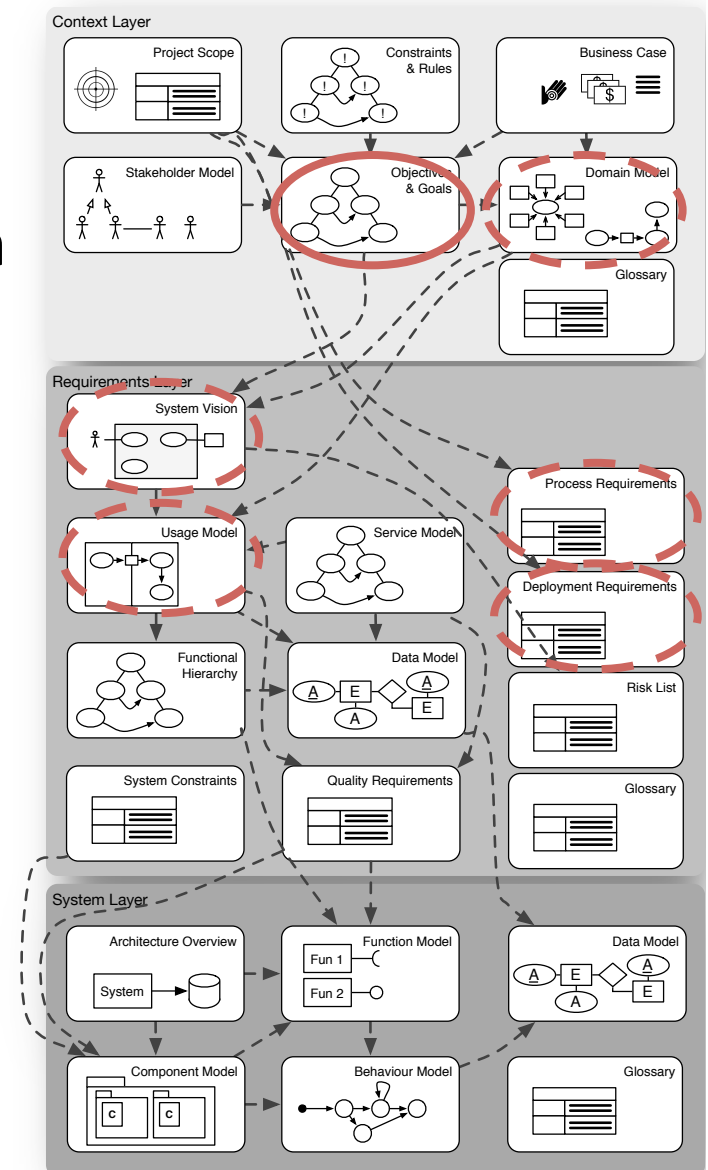
Eric Yu: Why Agent-oriented Requirements Engineering? Proc. 3rd REFSQ, 1997.

Goals and Constraints

- Motivation and overview
- Definition
- Types of goals
- Goal refinement and abstraction
- Goal models and usage for analysis and quality assurance
- Techniques for goal modeling
- Goal modeling with AMDiRE
- Typical problems when using goal models

Goals and other content items

- The **goal model is the basis** for
 - Early identification and resolution of conflicts
 - Rationale of a requirement
 - Modeling of the system behavior on different levels of abstraction
 - Domain Model
 - System Vision
 - Usage Model



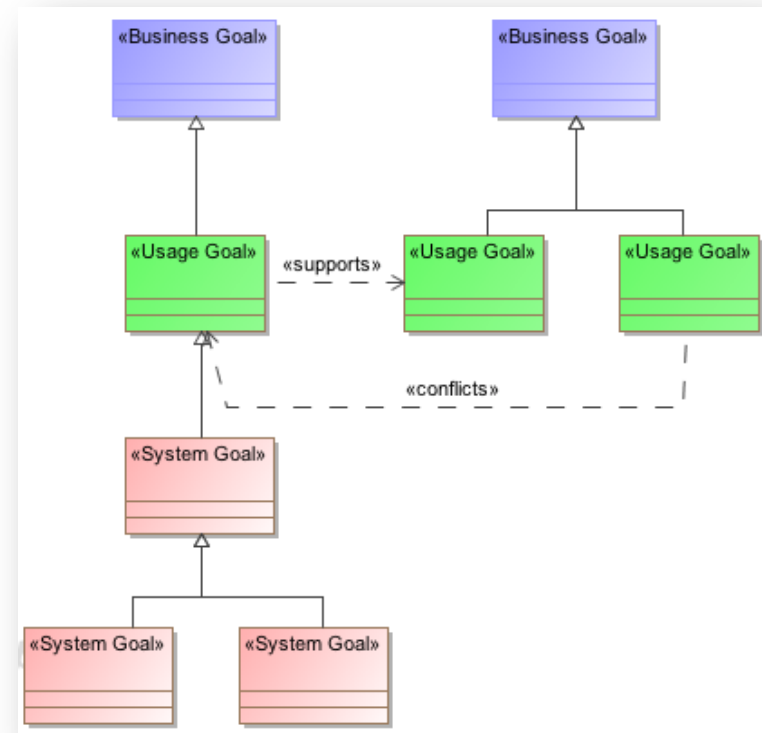
Goal categories and dependencies

Categories

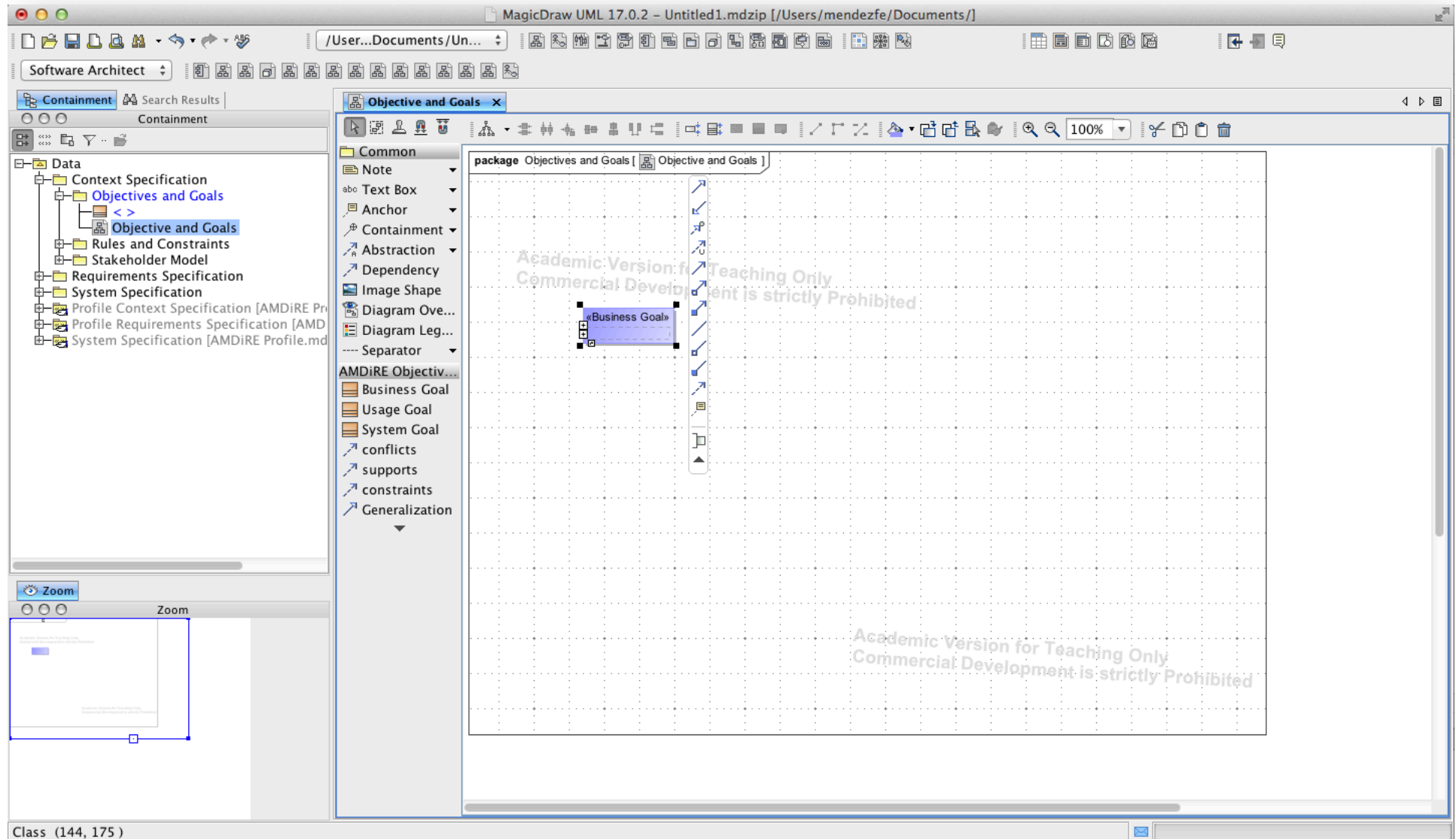
- **Business Goals:** all business-relevant (strategic) goals as well as goals with direct impact on the system or project.
- **Usage Goals:** direct relation to functional context and usage of the system (user perspective)
→ for behavior modeling
- **System Goals:** system-related goals that target system characteristics
→ to determine system characteristics

Dependencies (selection)

- **Decomposition** of goals into subgoals.
- **Supports:** A goal supports in achieving another goal.
- **Conflicts:** A goal is in conflict with another goal.



Elaborating a goal model



Problems in goal modeling

- Goals often intentional
 - Explain desired state but not how to reach it
 - Fuzzy, imprecise
 - Hard to determine when a goal is satisfied
- Goals are often in **conflict**
 - Optimization of one goal is in conflict with another one, e.g.:
 - *The system shall be of high quality*
 - *The system shall be cheap*
- **Abstraction levels & precision in differentiating between problem and solution**
 - General goal trees allow for a smooth transition between problem and solution.
 - Danger of veering off into architecture design too early.
 - **Solution orientation**

Exercise

Take your goals from the exercise last time for the car sharing system:

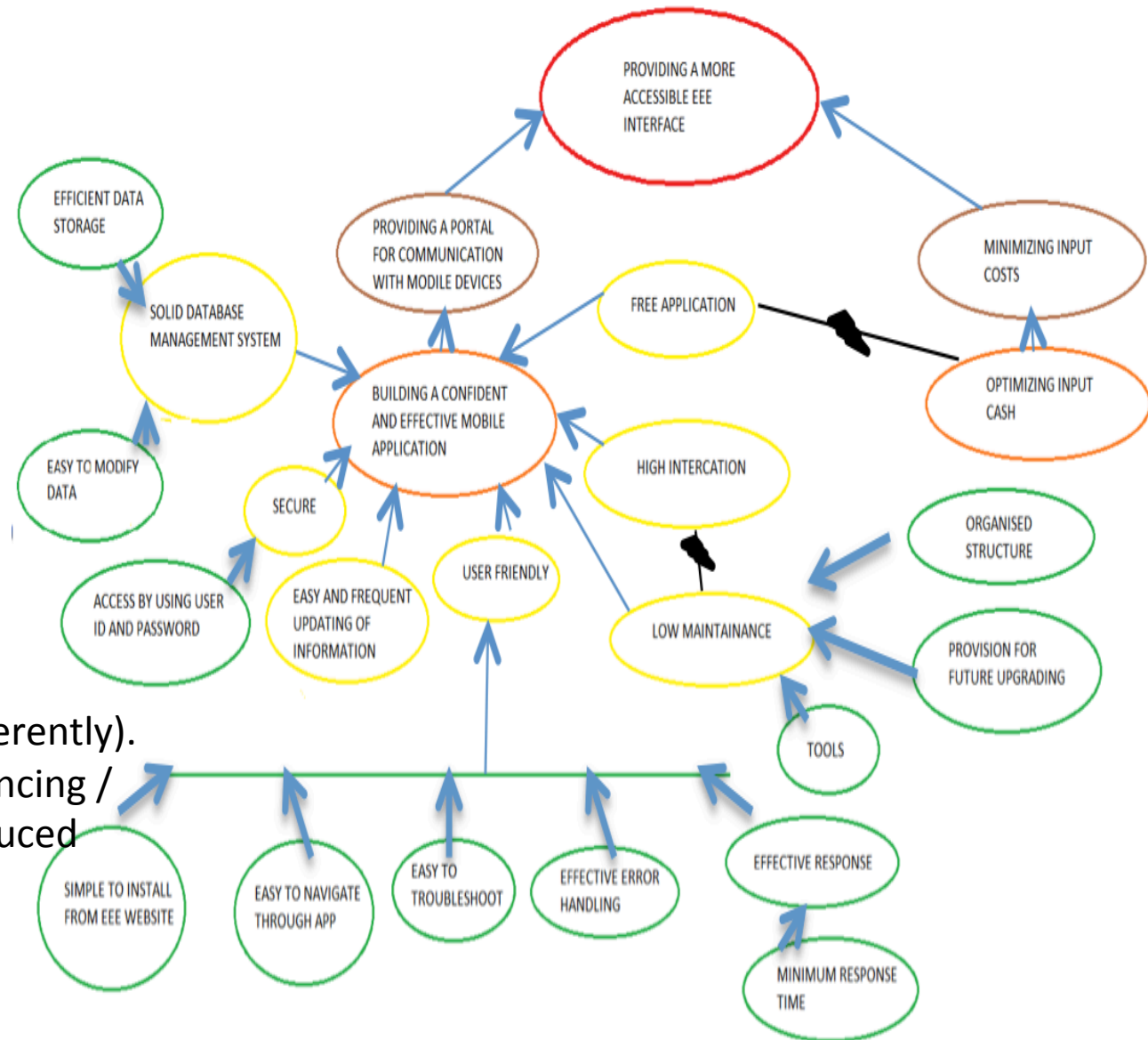
- Quality-related goals
- Optimization goals
- Behavioral goals
- Anti-goals

And re-classify them as Business Goals, Usage Goals, System Goals, and Constraints.

Which classification seems to be more useful to you and why?

Example 1

A tree-shaped hierarchy would be easier to read.
Missing: user features
(or should be phrased differently).
If we only talk about enhancing /
extending the system, reduced
features might be ok.

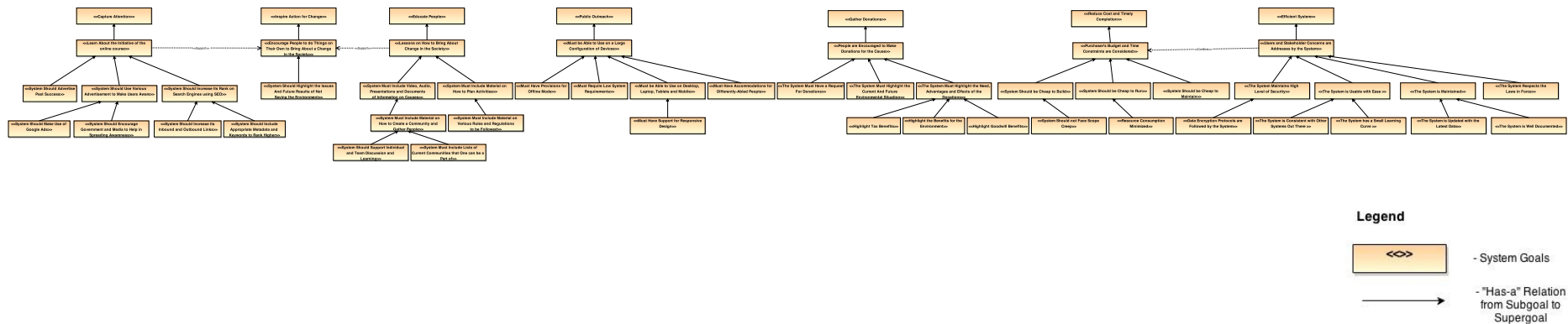


Above,

Red - business goals
Brown - system goals
Yellow and green - usage goals
Black lines - conflicts

Example 2 (PDF in Beach Board)

Figure 2: Goal Model³

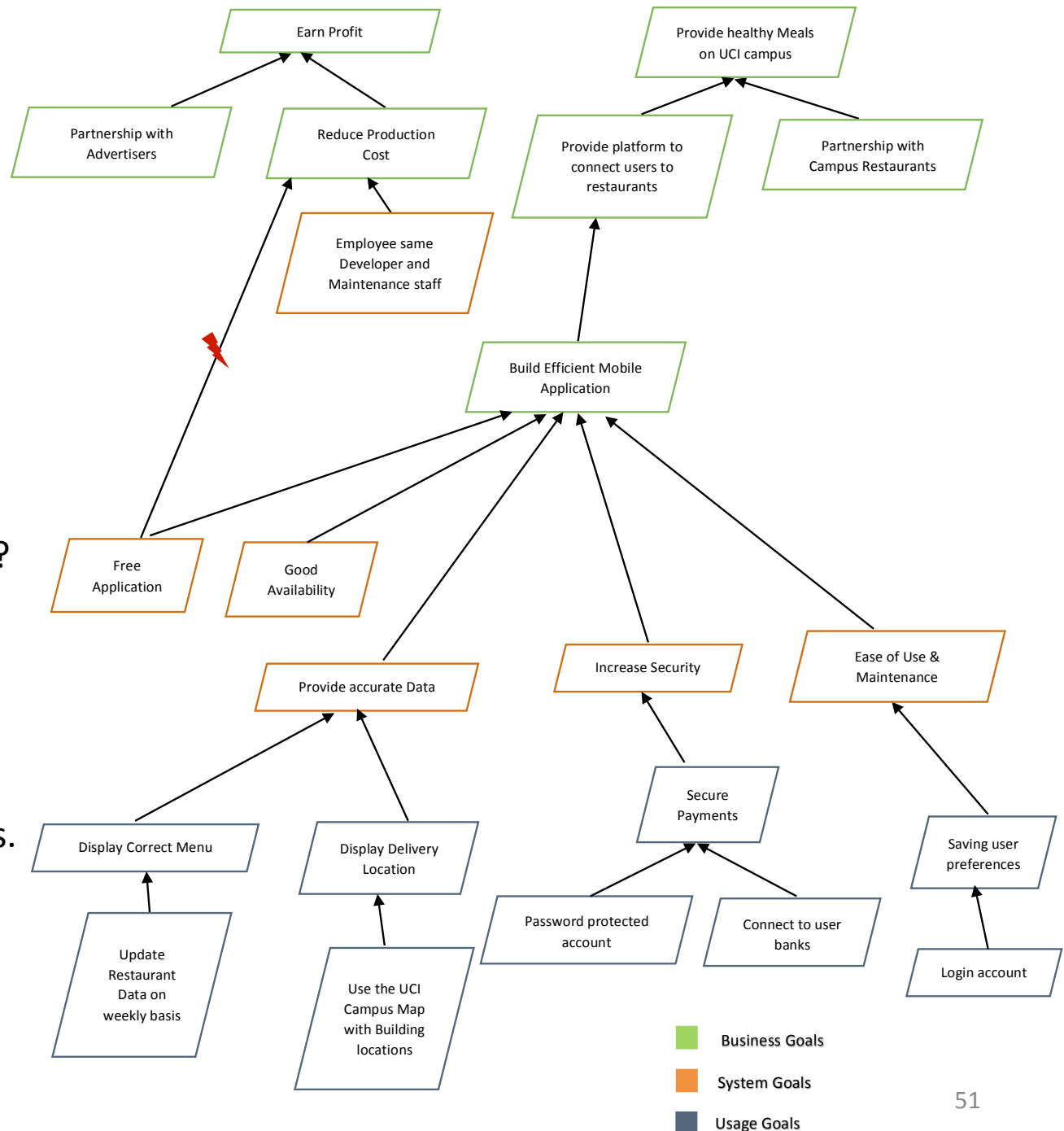


Try to put it on one page in a way I can still read it.

Example 3

Careful!
What's different in here
from the AMDiRE model?

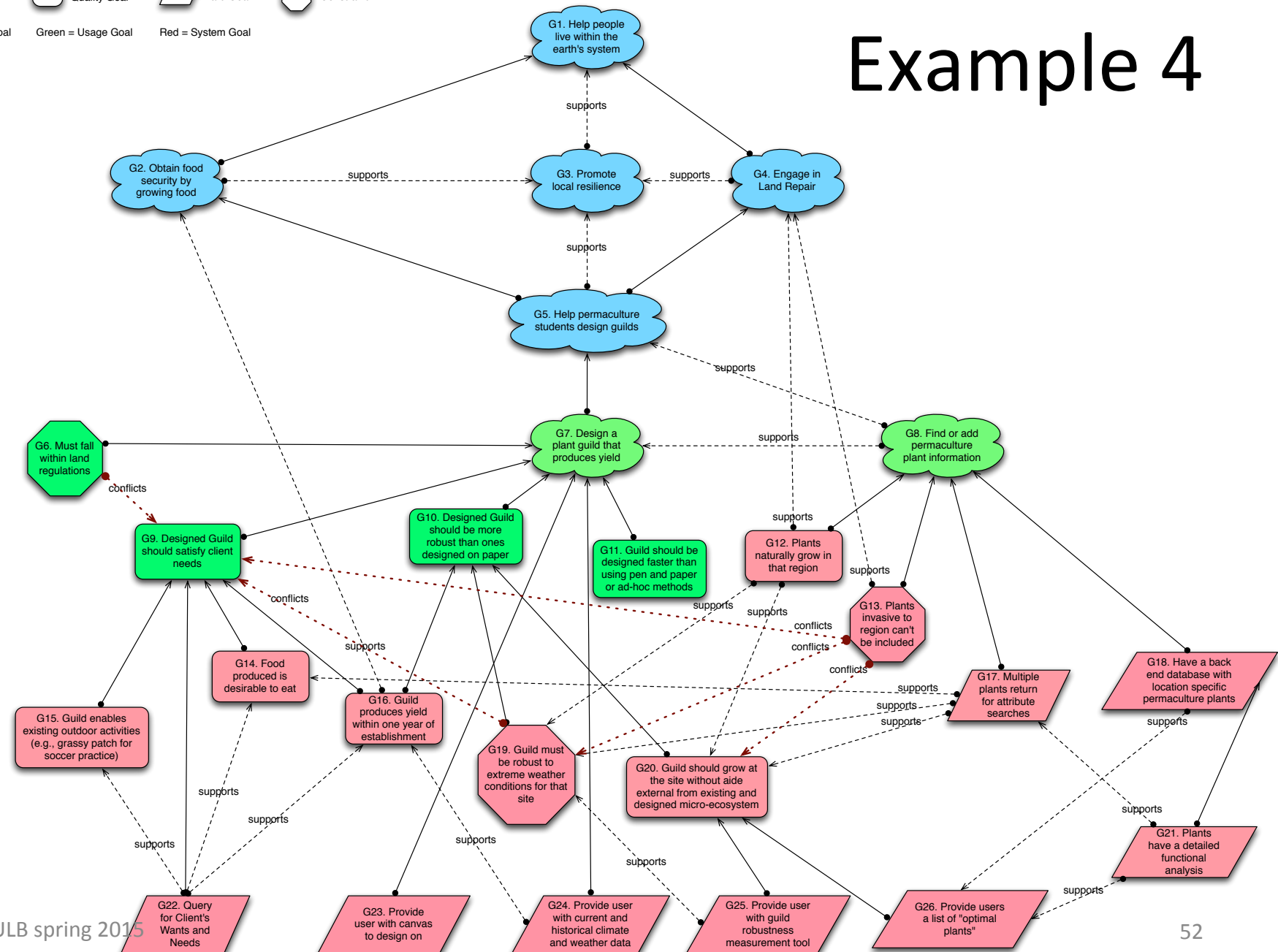
Try to refine Usage goals
from Business Goals.
System goals can be
derived from both
Business and Usage goals.



The Plant Guild Composer: Goal Model



Example 4



Exercise

Let's make a goal model for the car sharing system:

- Business goals
- Usage goals
- System goals

Then let's define the relations between them:

- Subgoal
- Supports
- Conflicts

Assignment

Make a goal model for your project system.
Let's take a look at the assignment description.