Goal Modeling and GRL

Gregor v. Bochmann, University of Ottawa

Based on Powerpoint slides by Gunter Mussbacher (2009)
with material from Amyot

First part:

Goal Modeling

This part is largely based on the book
“Requirements Engineering”
by Axel van Lamsweerde

Different types of statements

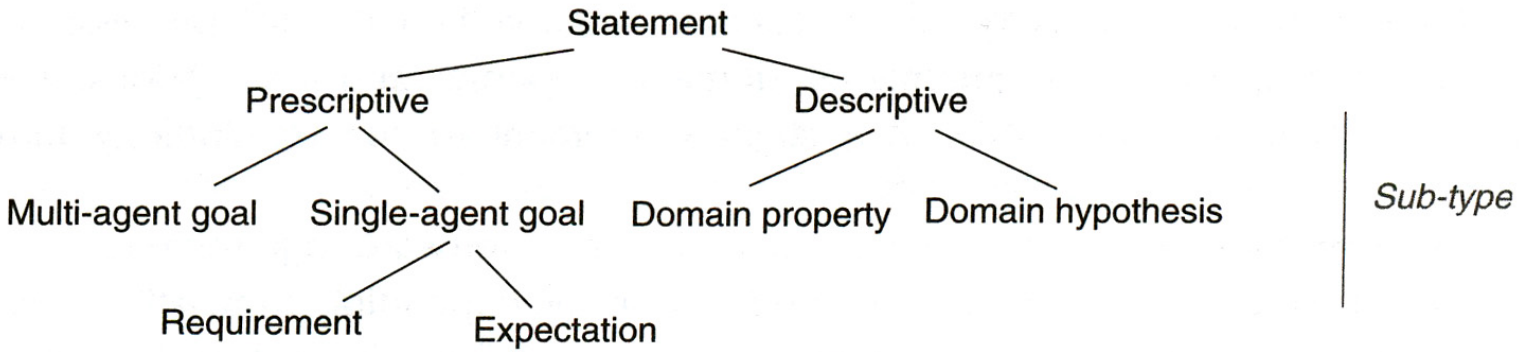
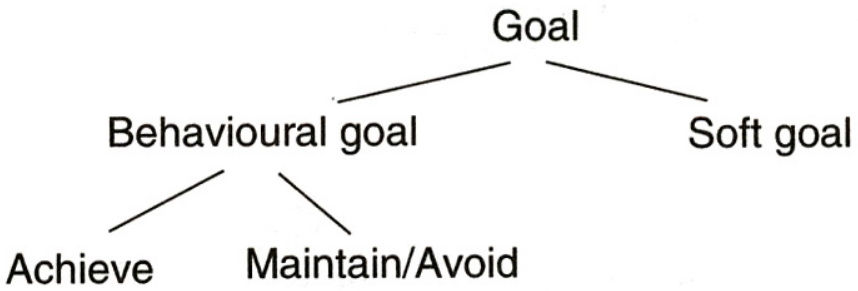


Figure 7.1 *Statement typology with goals*

- Descriptive: describes some (existing) situation
- Prescriptive: describes some situation that is desired to be true
- Goal: "a prescriptive statement of intent that the system should satisfy through the cooperation of its agents"
  - Goal satisfaction may involve a variety of agents, such as actors in the system's environment, as well as the system as a whole or its components
- Requirement: a goal under the responsibility of a single agent (the system-to-be or a component of it)
- Expectation: a goal under the responsibility of a single agent in the environment of the system-to-be. Note: this is an assumption that the system can make

Different types of goals



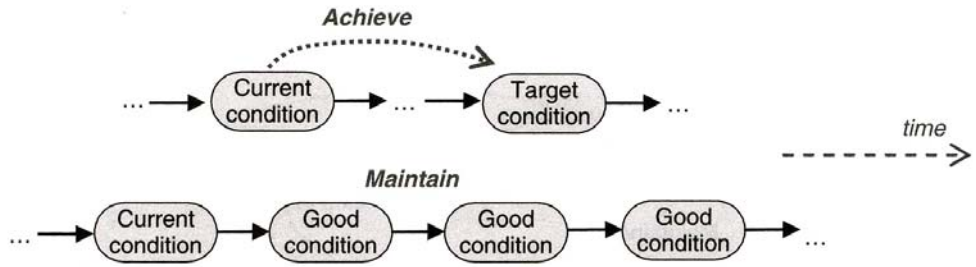


Figure 7.4 *Behavioural goals: Achieve and Maintain goals*

Figure 7.2 *A taxonomy of goal types*

- Behavioral goal: establishment of goal can be checked
  - Describes intended behavior declaratively
  - Implicitly defines a maximal set of admissible behaviors
    - Achieve: points to future (like "eventually" operator in Temporal Logic)
    - Maintain/Avoid: states property that always holds (like "always" operator)
- Soft-Goal: are more or less fulfilled by different alternatives of (external) design – often difficult to quantify – one says, some alternative may "satisfice" the goal

Different categories of goals (requirements)

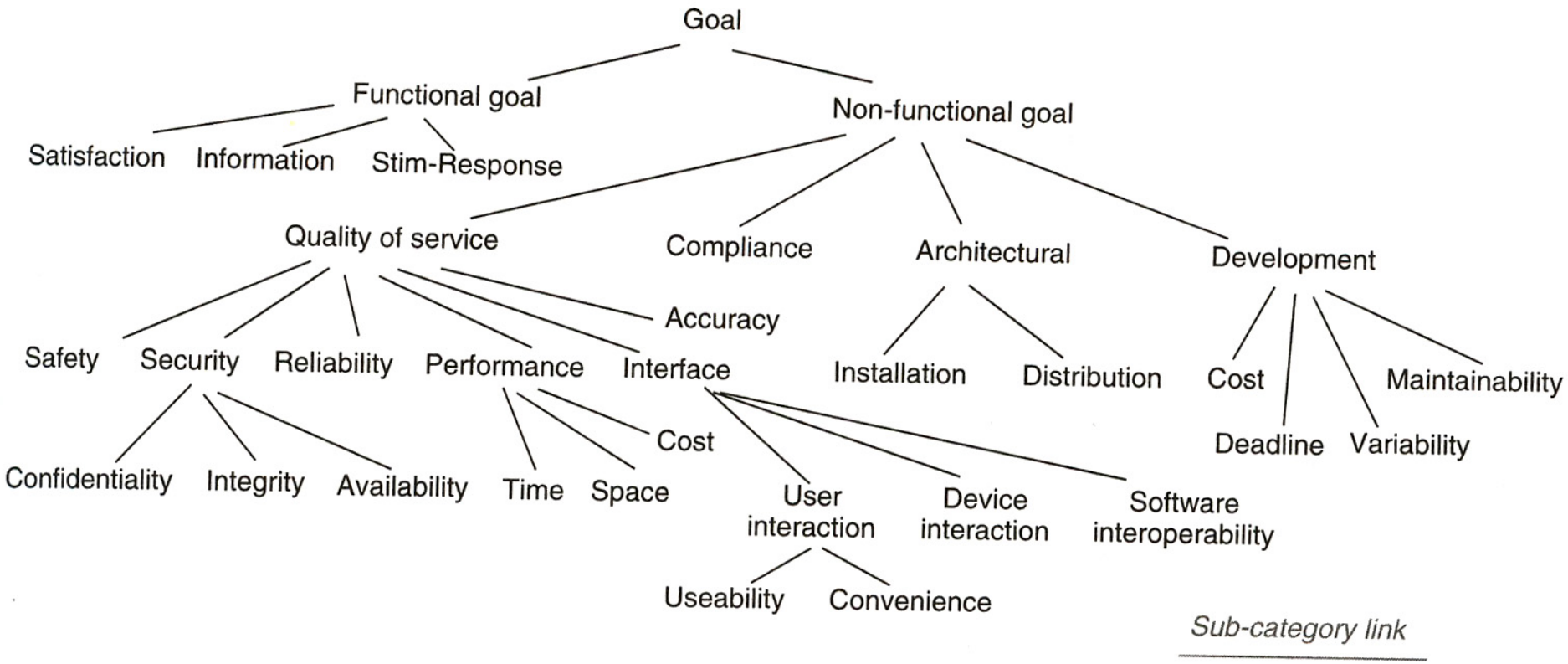
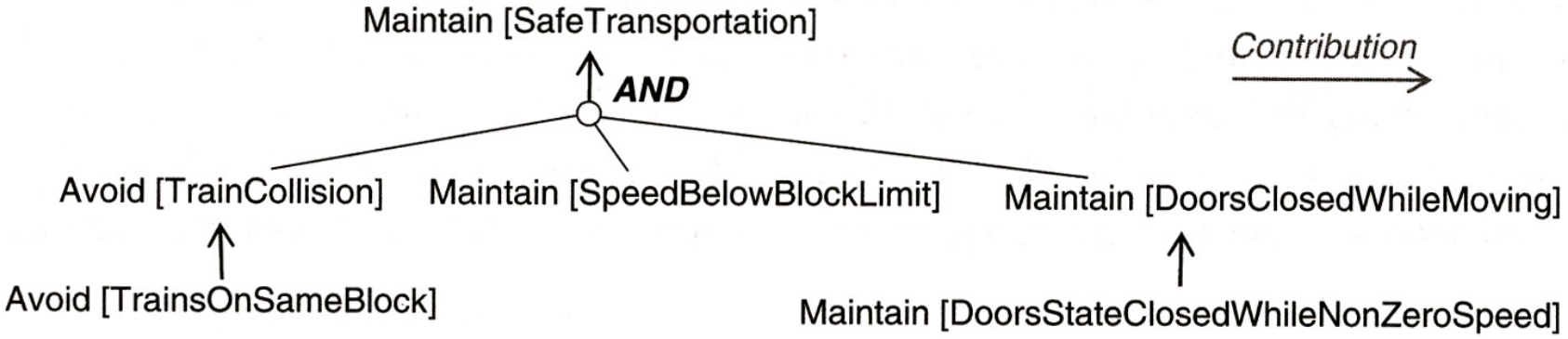


Figure 7.5 *Goal categories*

This is the same as the classification of requirements into functional and non-functional (with all its sub-categories)

Goal refinement

- Goal refinement: expressing how a more abstract goal can be established by a set of more low-level goals – AND and OR refinement



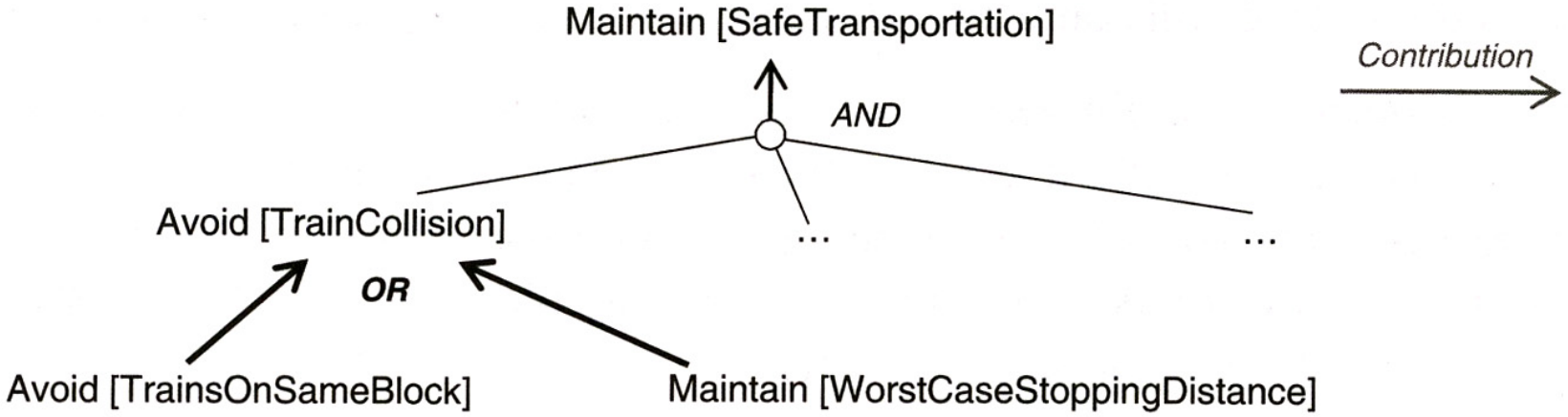
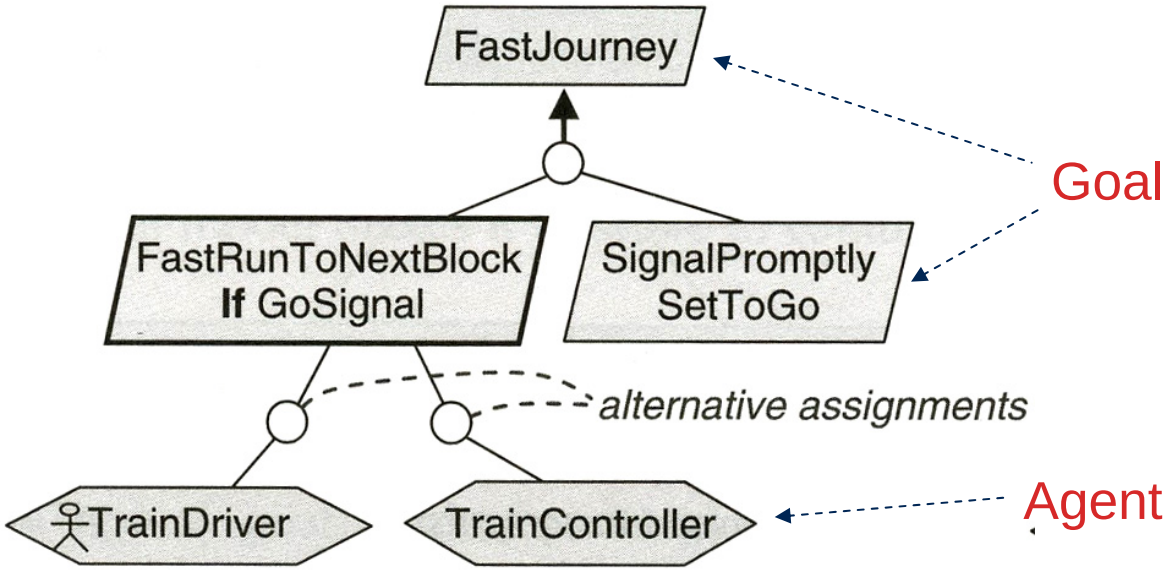


Figure 7.7 *Goal OR-refinement, alternative options, and system versions*

Goal - responsibility assignment

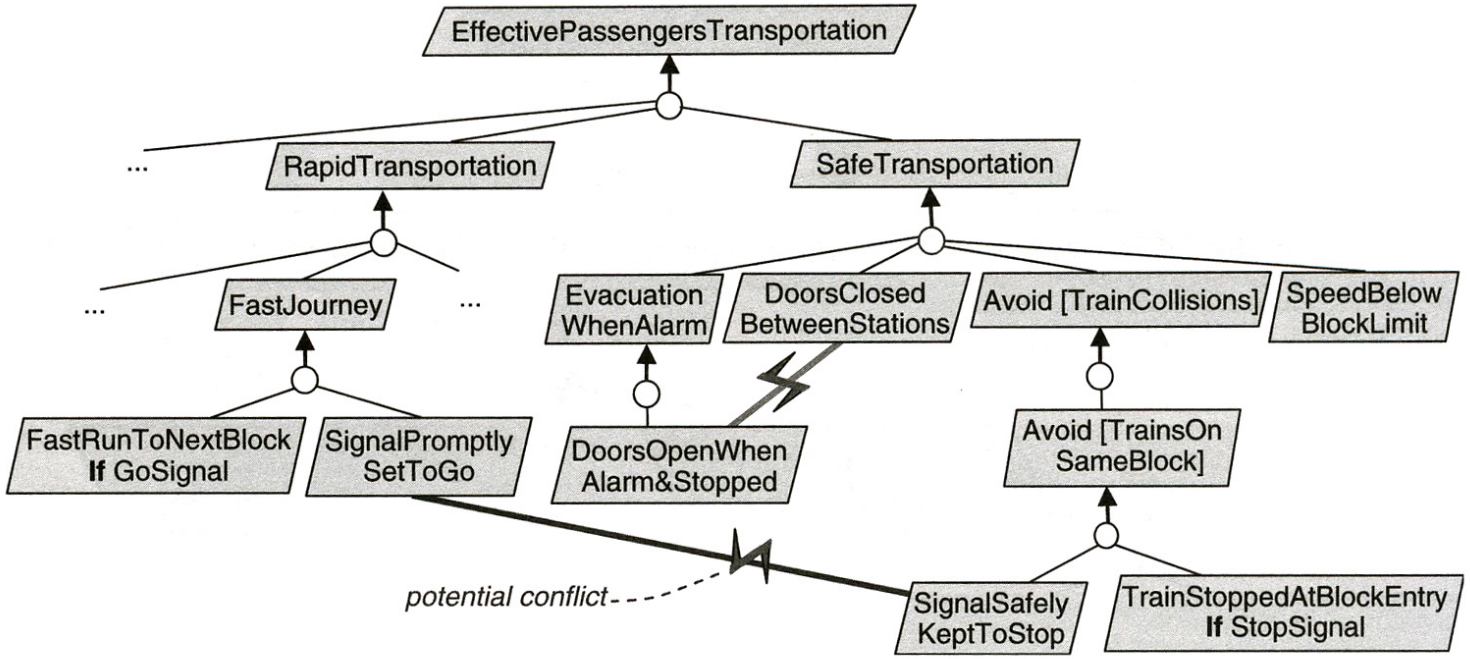
- Responsibility assignment: who are the responsible agents ?
  - Note: For a high-level goal, there are often several agents involved.

- Example: only one agent - but two alternatives

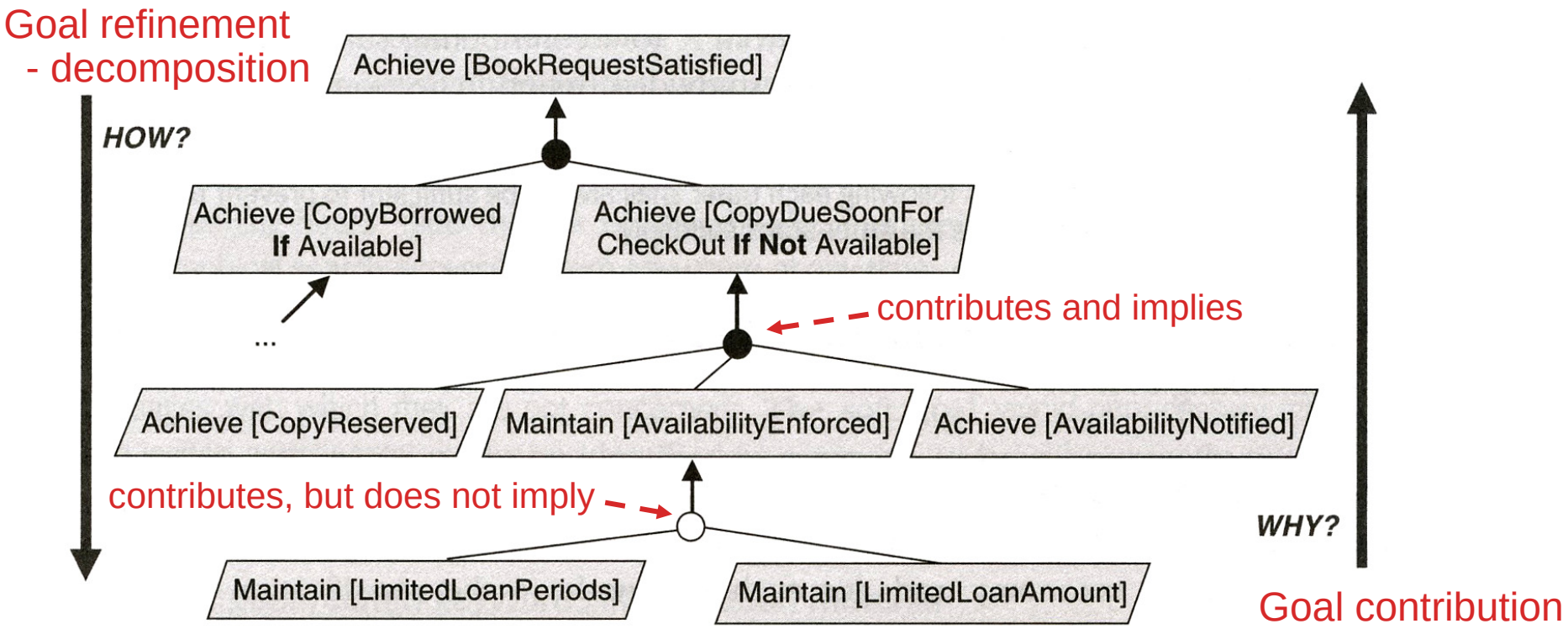


Risks – obstacles - conflicts

- Risk: “uncertain factor whose occurrence may result in the loss of attainment of some corresponding objective” - “goal negation”
- Obstacle to a goal: a pre-condition for the non-satisfaction of the statement – that is, if the obstacle is true then the goal cannot be satisfied
- Conflict between several goals/requirements: conflicts between requirements are often due to conflicts between the underlying goals, which may belong to different stakeholders – need for conflict resolution with stakeholders



Reasoning about goals



- For precisely defined behavioral goals and corresponding refinement tree, one can present proofs of correctness for the reasoning.
- Rationale: reasoning behind some (external) design choice or the statement of some goal contribution

Example: a goal decomposition that can be verified

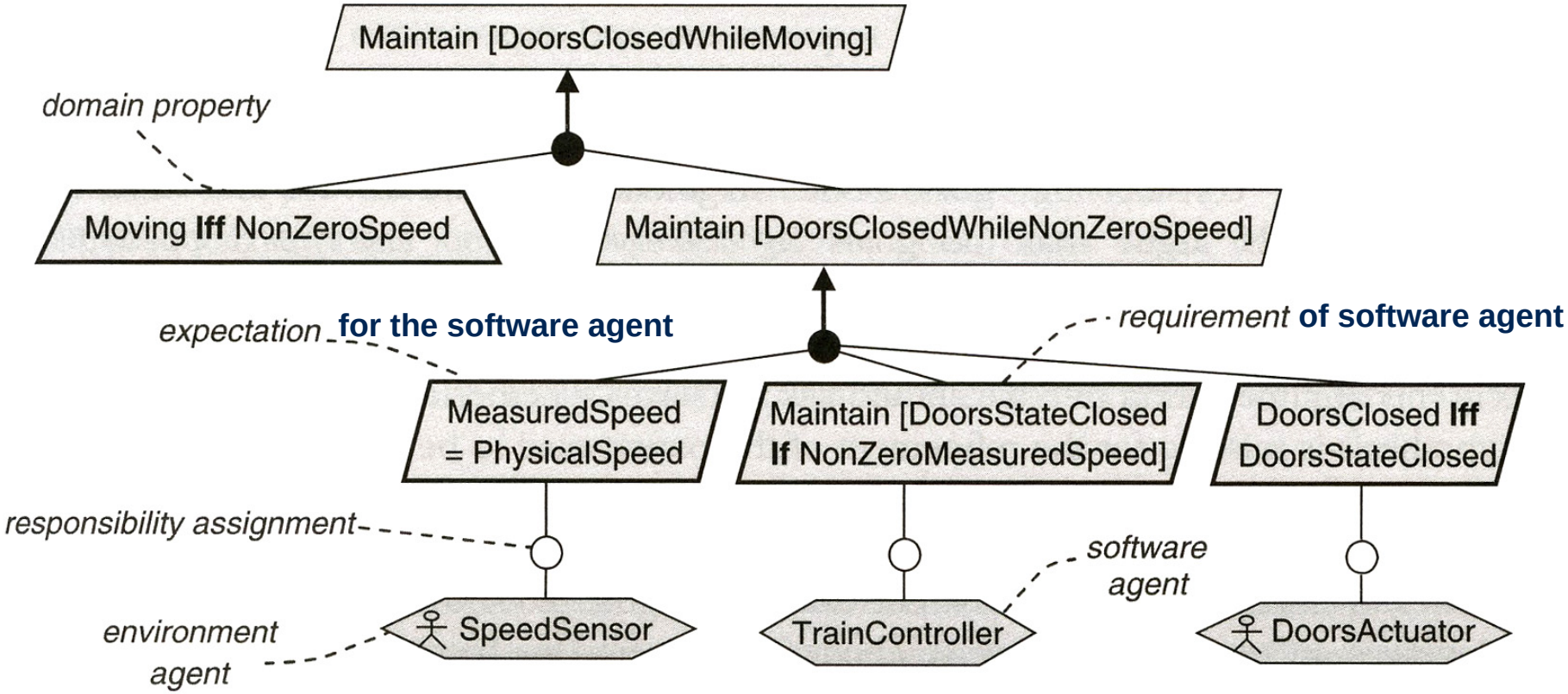


Figure 8.4 *Leaf nodes in an AND-refinement tree*

Example: a faulty assumption in the rationale

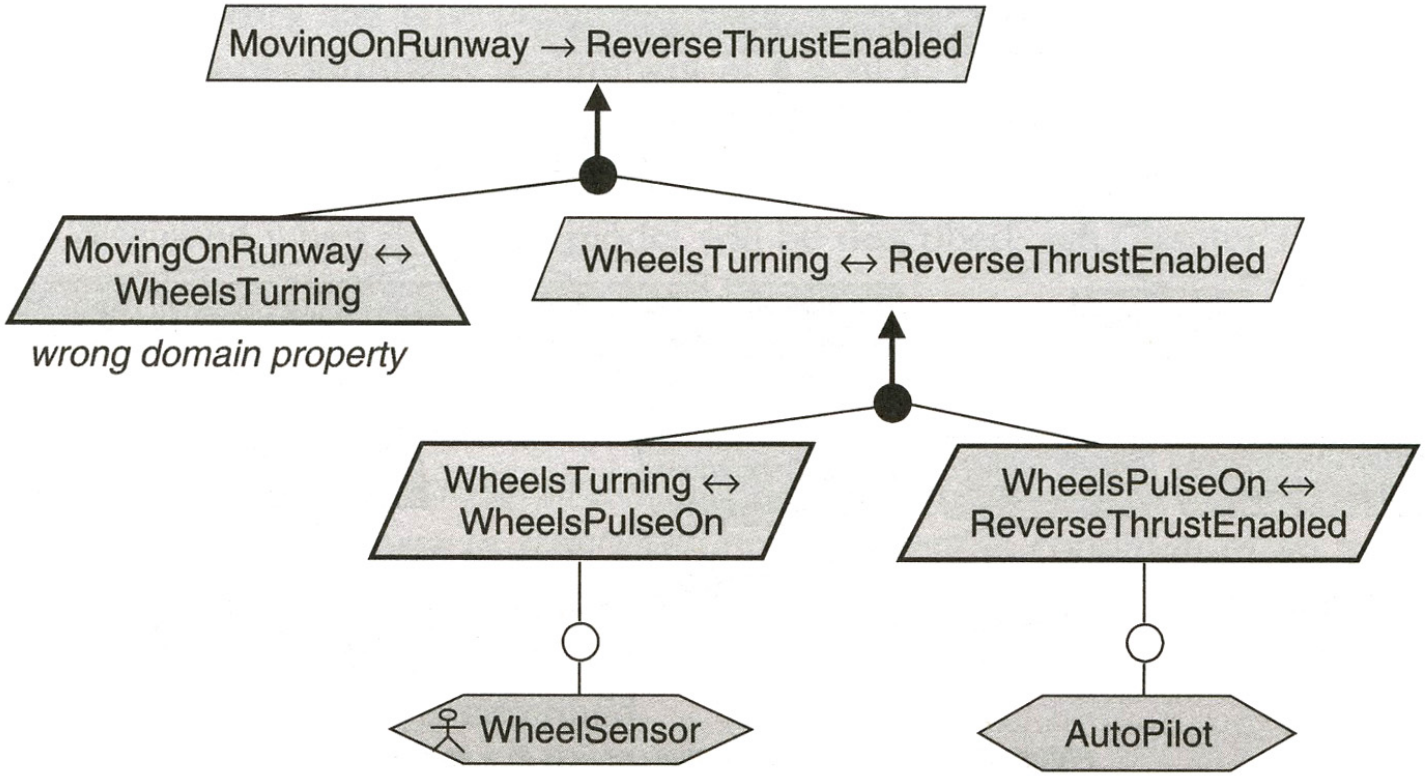


Figure 8.6 *Correct refinement based on wrong domain property for the Airbus A320 braking logic*

Example: alternative goal refinements

- This means, different alternatives for the (external) design

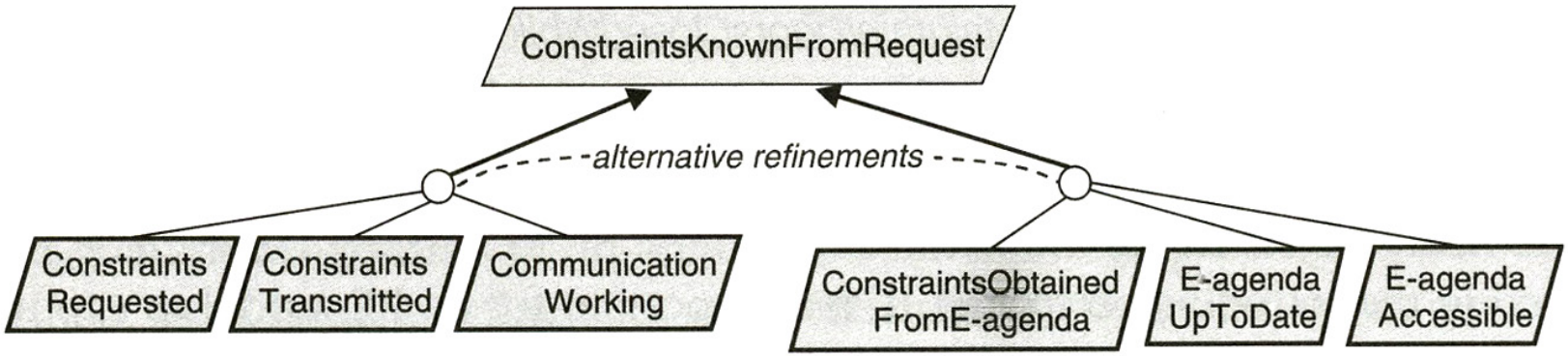
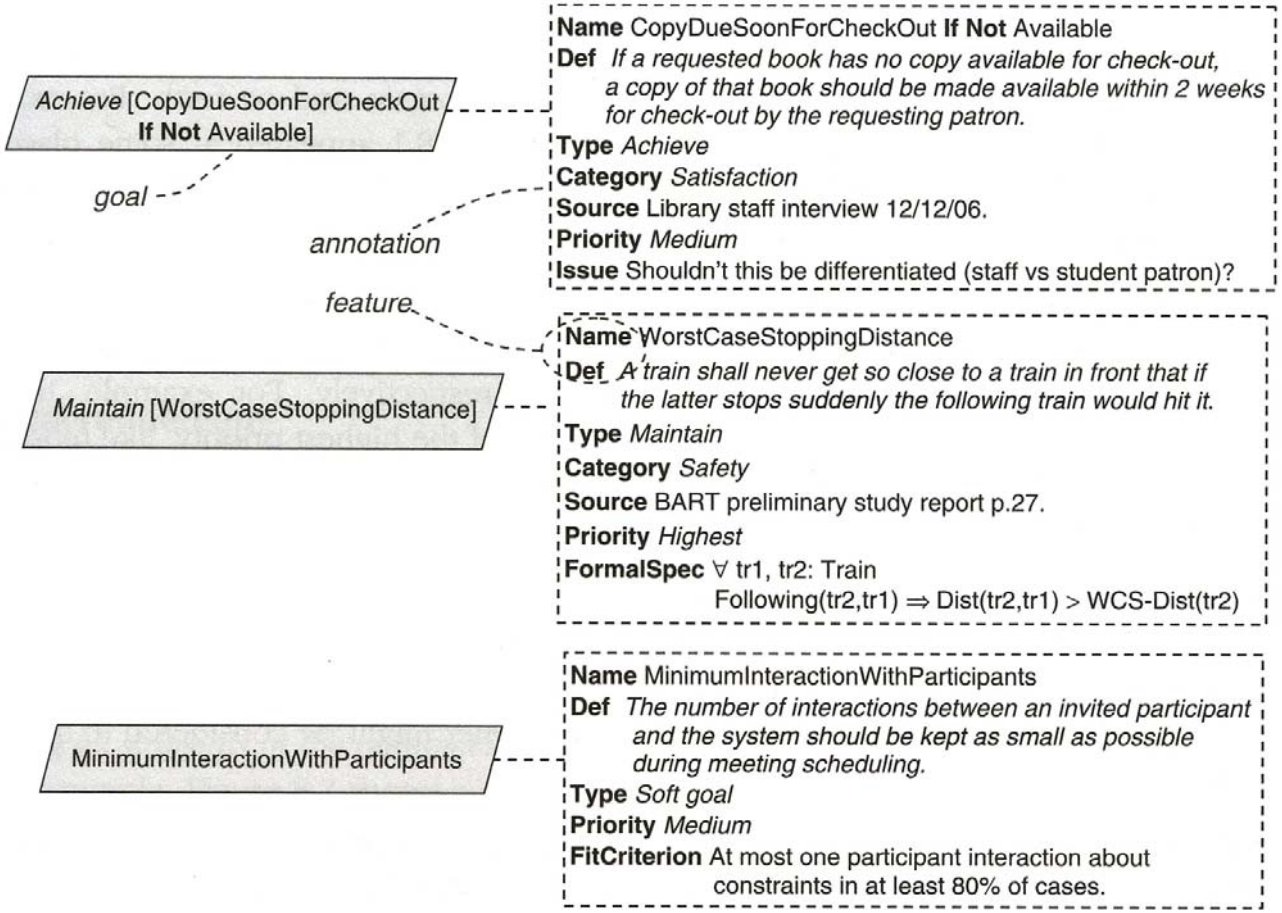


Figure 8.8 *Alternative goal refinements in the meeting scheduling system*

- Given some other soft goal (not mentioned above), this goal may be satisficed to different degrees by the two alternatives.
- This may become the rationale for selecting one of the alternatives.

Templates / notations for defining goals and contributions



- There are many notations for goals and goal contributions
  - Notation from the book by Lamsweerde
  - i* = istar (developed at the University of Toronto) and GRL (see below)

Goal contributions – tracability (examples of notations)

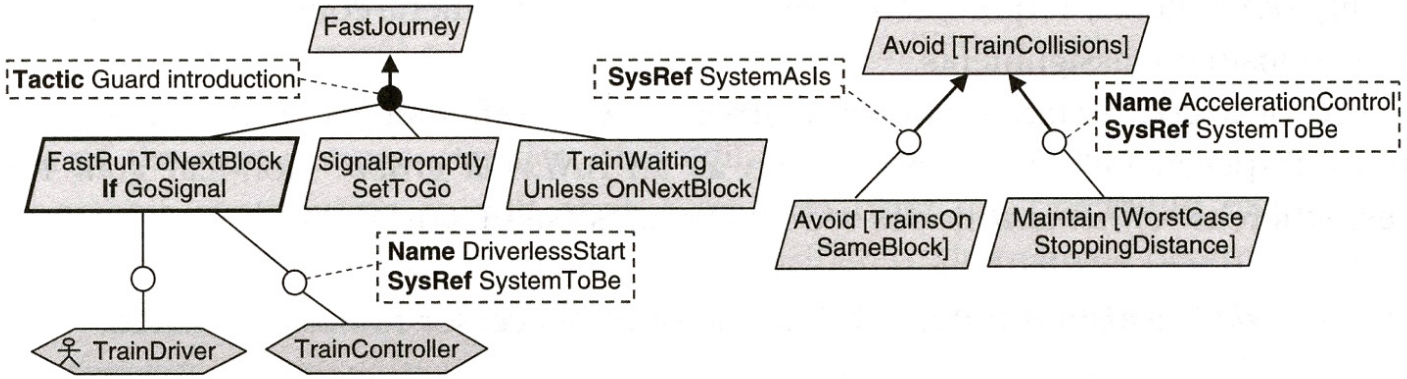


Figure 8.12 *Annotating refinements and assignments*

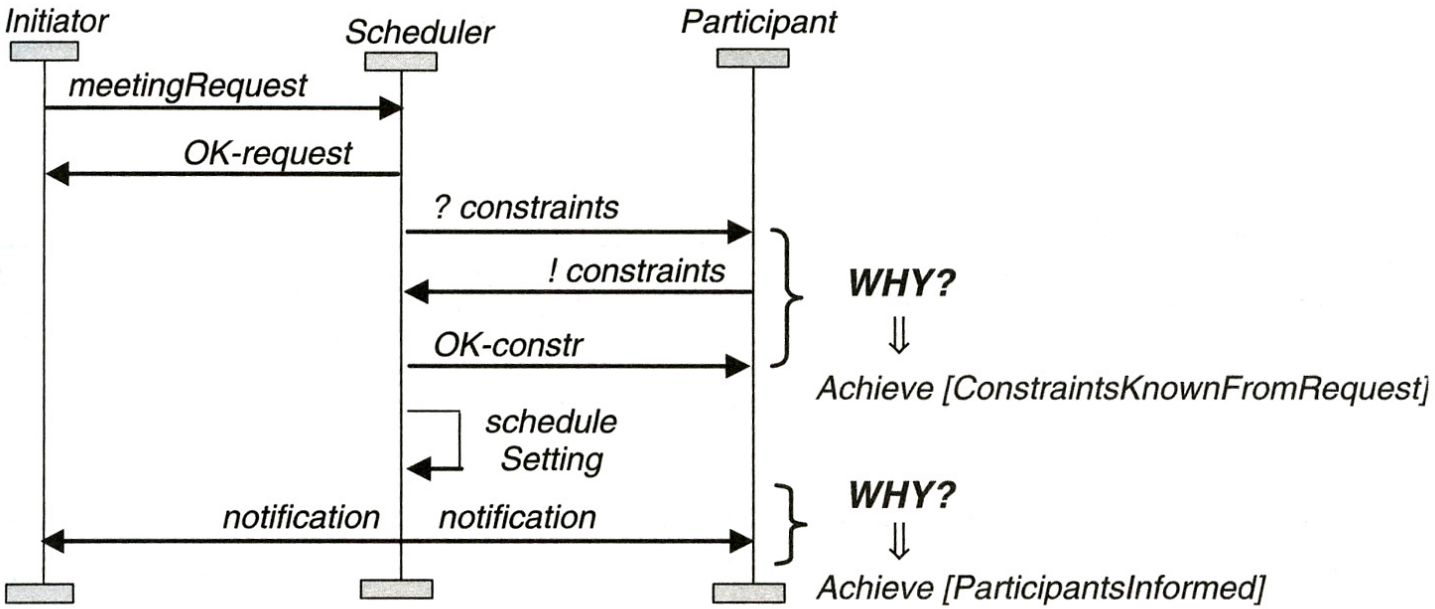


Figure 8.14 *Goal identification from WHY questions on scenario episodes*

The central role of goals in requirements engineering

- Goal refinement provides a natural mechanism for structuring complex specifications at different levels of concern.
- Goals provide the rationale for requirements.
- Goals drive the identification of requirements to support them.
- Goals provide a richer structure for satisfaction arguments.
- Goals provide a basis for showing the alignment of the system-to-be with the organization's strategic objectives.
- Goals provide a precise criterion for requirements completeness.
- Goals provide a precise criterion for requirements pertinence.
- Goals provide a natural way of structuring the RD.
- Goals provide anchors for risk analysis.
- Goals provide the roots for managing conflicts among requirements.
- Goals provide a criterion for delimiting the scope of the system.
- Goals support the analysis of dependencies among agents.
- Goals provide a basis for reasoning about alternative options.
- Goals support traceability management.
- Goals provide essential information for evolution support.

Another definition of "goal"

- Goal: high level objective of the business, organization or system
  - A requirement specifies how a goal should be accomplished by a proposed system
- Operationalization: the process of defining a goal with enough detail so that its sub-goals have an operational definition.
- Decomposition: the process of subdividing a set of goals into a logical sub-grouping so that system requirements can be more easily understood, defined and specified.
- Obstacles: behaviours or other goals that prevent or block the achievement of a given goal.
  - Abstracting and identifying goal obstacles allows one to consider the possible ways for goals to fail and anticipate exception cases.

Source: A. Antón

Comments* on previous slide

- This alternative definition of “goal” appears to be less useful.
- It seems to correspond to what Lamsweerde called SoftGoal
- Example of SoftGoals and their use for choosing between alternative (external) designs: Consider an ATM terminal

Question: Alternative Authentication Mechanisms?
References: Service: Authenticate (this is a Goal)

	Criteria 1: ATM Unit Cost	**Criteria 2:** Privacy
Option 1: Account number	+	–
Option 2: Fingerprint reader	–	+
Option 3: Smart Card + PIN	+	+

Criteria 1 and 2 are SoftGoals

* Source: G.v.B.

Introduction to the

Goal-Oriented Requirements Language

(GRL)

GRL Overview (1)

- The Goal-oriented Requirement Language (GRL)
  - Graphical notation
  - Connects requirements to business objectives
  - Allows reasoning about (non-functional) requirements
  - Is based on i* (concepts / syntax) and the NFR Framework (evaluation mechanism)

- GRL models the "why" aspect
  - Model goals and other intentional concepts
  - Little or no operational details
  - Supports goal and trade-off analysis and evaluations

GRL Overview (2)

GRL is used to …

- Visually describe business goals, objectives, stakeholders' priorities, alternative solutions, rationale, and decisions
- Decompose high-level goals into alternative solutions called tasks (this process is called operationalization)
- Model positive & negative influences of goals and tasks on each other
- Capture dependencies between actors (i.e., stakeholders)
- Reason about alternatives and trade-offs

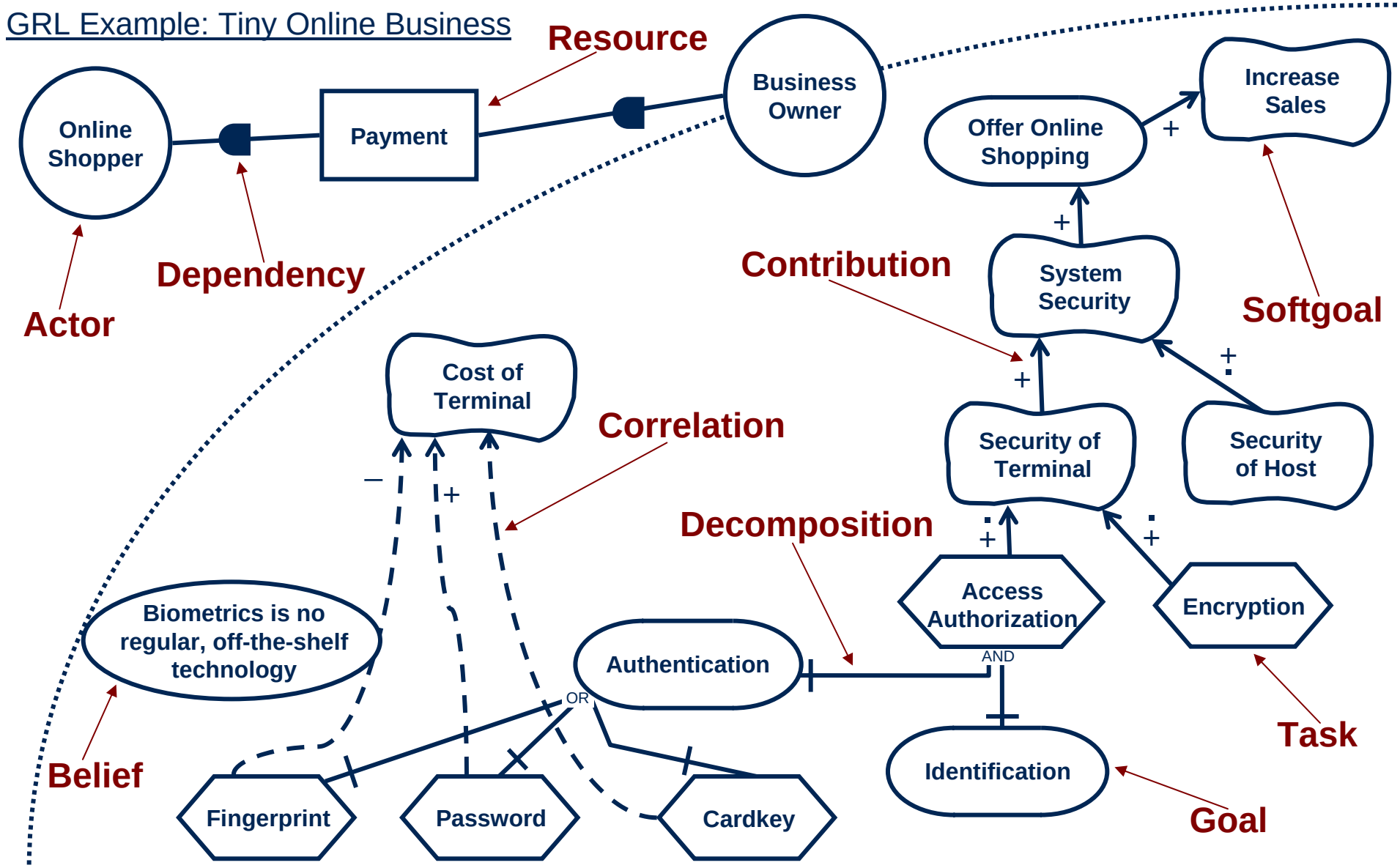
In essence …

GRL can be used for what we discussed above using the notation from Lamsweerde's book. GRL uses a different notation and has some additional concepts.

GRL is mainly designed for SoftGoals with their fuzzy satisfaction criteria. Not intended for the verification of behavioral goals, as in some of the examples above.

There is a tool, called jUCMNav, that supports this language.

GRL Notation – an example



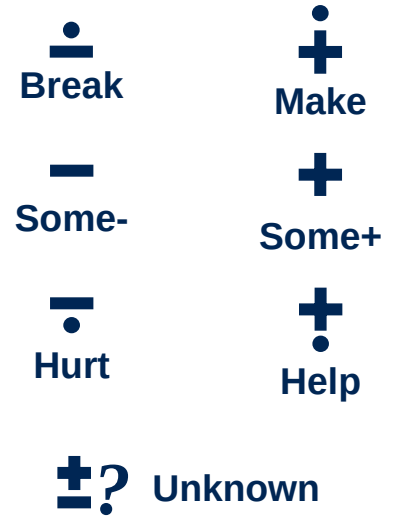
GRL Concepts

- Concepts already discussed in Lamsweerde's notation
  - Goal, Softgoal, contributions including AND, OR, also XOR relationships (the GRL correlation is similar)
    - Note: Achievement of softgoal is qualifiable but not measurable; it is quantifiable for goals (Softgoals are often non-functional, goals functional)
  - Actor : appears to be a subtype of Agent
  - Belief : appears to be the same as a Rationale
- Other concepts:
  - Task: a proposed solution that achieves a goal or satisfices a softgoal
    - This appears to be similar to the concept of agent responsible for realizing some goal
  - Dependency: An actor (the depender) depends on another actor (the dependee) for something (the dependum), e.g. the business owner depends on the online shopper for payment (the dependum is optional)
  - Resource: used in dependencies as dependum

GRL Notation (2)

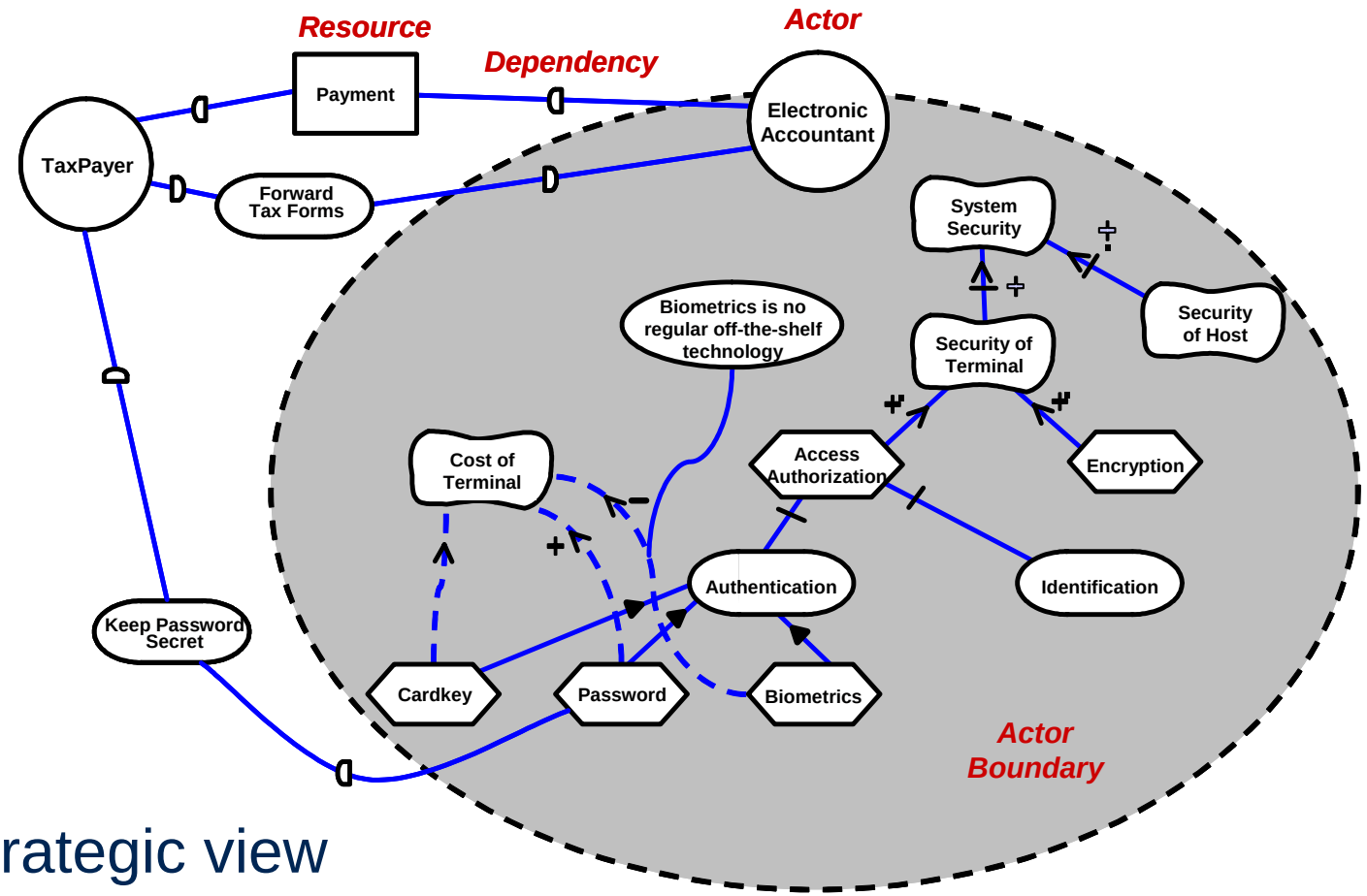
- Contribution and Correlation Links
  - Contribution describes desired impact, correlation shows side effects
  - Qualitative or quantitative contribution types are used for these links
- Note: In GRL, contributions can be negative, that is, the contributor (sub-goal or task) may be an obstacle.

GRL Contributions Types: (qualitative)



GRL Notation (3)

- GRL graphs can be allocated to actors
- Dependencies can be defined between actors together with intermediate resources or other elements



- Provides a strategic view

Note: this is an i* model and therefore the syntax is slightly different

Why GRL?

These are essentially arguments that were already given when we discussed goals above.

- Goals become an important driver for requirements elaboration – yet, stakeholders goals and objectives are complex and will conflict…
- GRL expresses and clarifies tentative, ill-defined, and ambiguous requirements
  - Supports argumentation, negotiation, conflict detection & resolution, and in general decisions
  - Captures decision rationale and criteria (documentation!)
- GRL identifies alternative requirements and alternative system boundaries
- GRL provides clear traceability from strategic objectives to technical requirements
- GRL allows reuse of stable higher-level goals when the system evolves
- **There is nothing like this in UML…**

GRL – Strategies and Evaluation Mechanism (1)

- GRL allows a particular configuration of intentional elements to be defined in a strategy (i.e., one possible solution)
  - Captures the initial, user-defined satisfaction levels for these elements separately from the GRL graphs
  - Strategies can be compared with each other for trade-off analyses
- In order to analyze the goal model and compare solutions with each other, jUCMNav's customizable evaluation mechanism executes the strategies
  - Propagating satisfaction levels to the other elements and to actors shows impact of proposed solution on high level goals for each stakeholder
  - Propagation starts at user-defined satisfaction levels of intentional elements (usually bottom-up)

GRL – Strategies and Evaluation Mechanism (2)

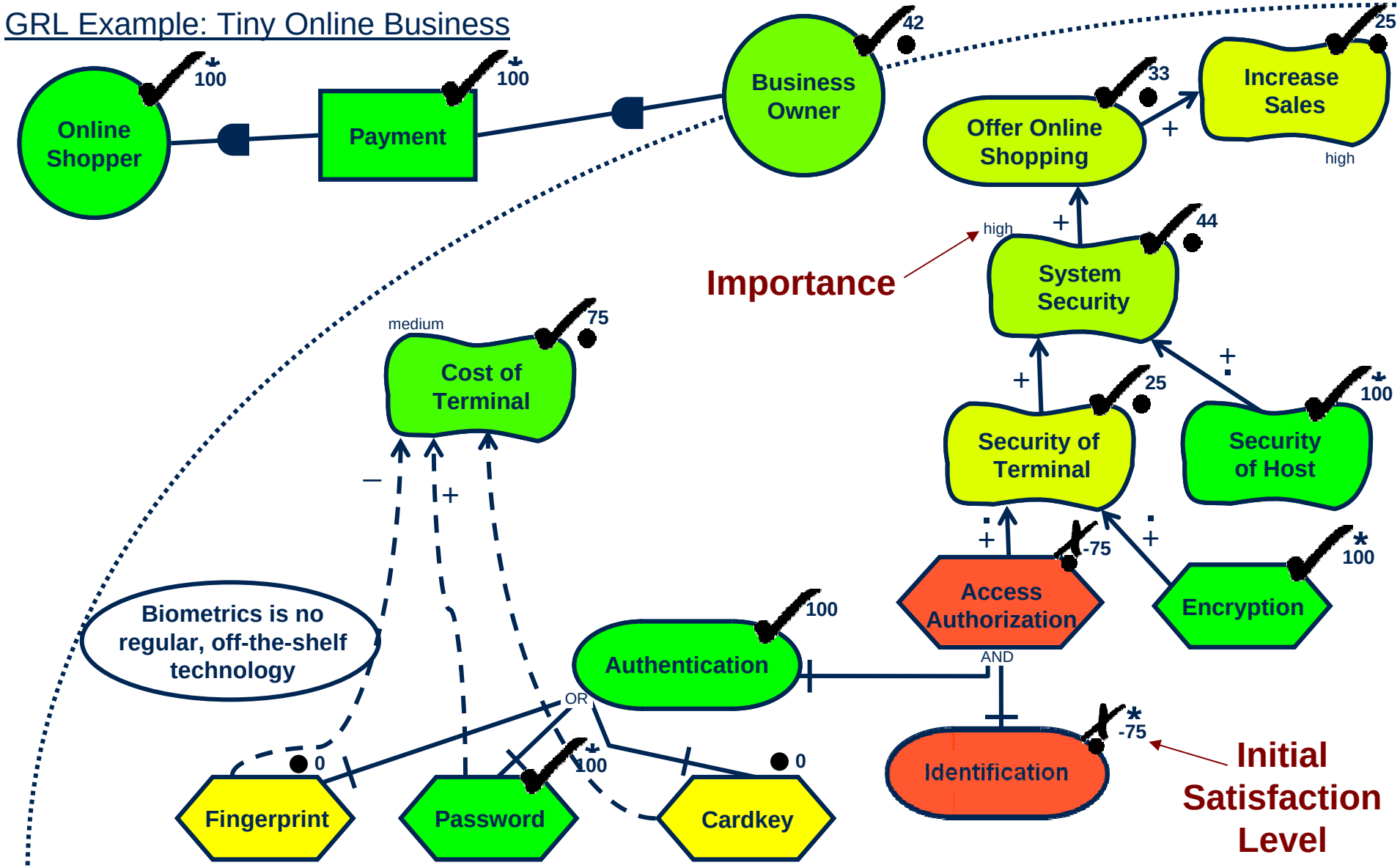
- Evaluations of GRL graphs show the impact of qualitative decisions on high-level softgoals
- Evaluation mechanism takes into consideration
  - Initial satisfaction levels of children (intentional elements)
  - Links, types of these links, and contribution/decomposition types
  - Importance defined for intentional elements
- More complete than simple pros/cons tables or criteria evaluation matrices
- For details, see Chapter 11.1 and Appendix II of the Z.151 standard
  - Standard provides minimum requirements

GRL – Qualitative or Quantitative Approach

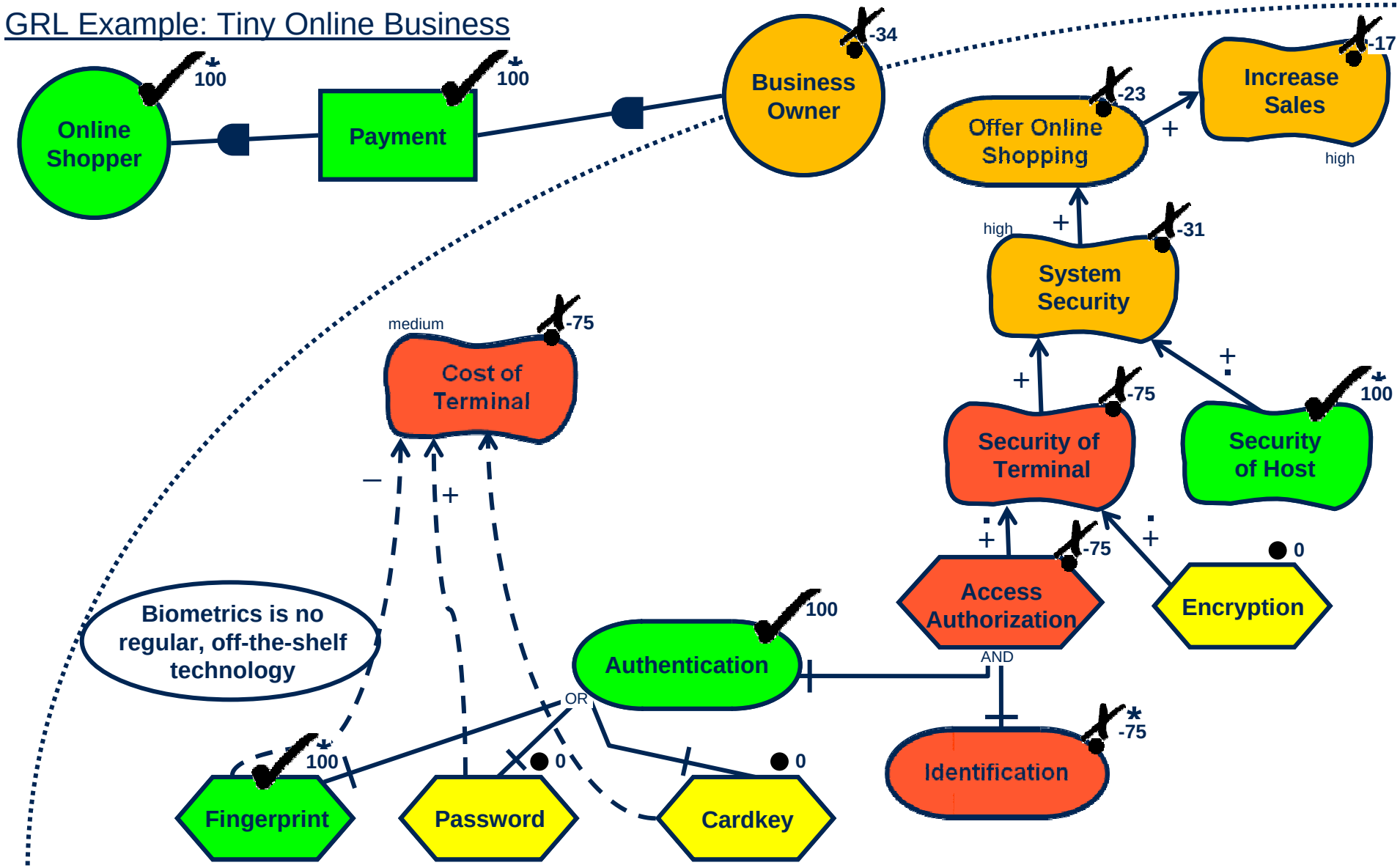
- Qualitative Approach
  - Contribution types: from Make to Break
  - Importance: High, Medium, Low, or None
  - Qualitative satisfaction levels
- Quantitative Approach
  - Contribution types: [-100, 100]
  - Importance: [0, 100]
  - Quantitative satisfaction levels: [-100, 100]
- Hybrid Approach is also possible
  - Qualitative contribution types
  - Quantitative importance
  - Quantitative satisfaction levels

GRL Satisfaction Levels: (qualitative)

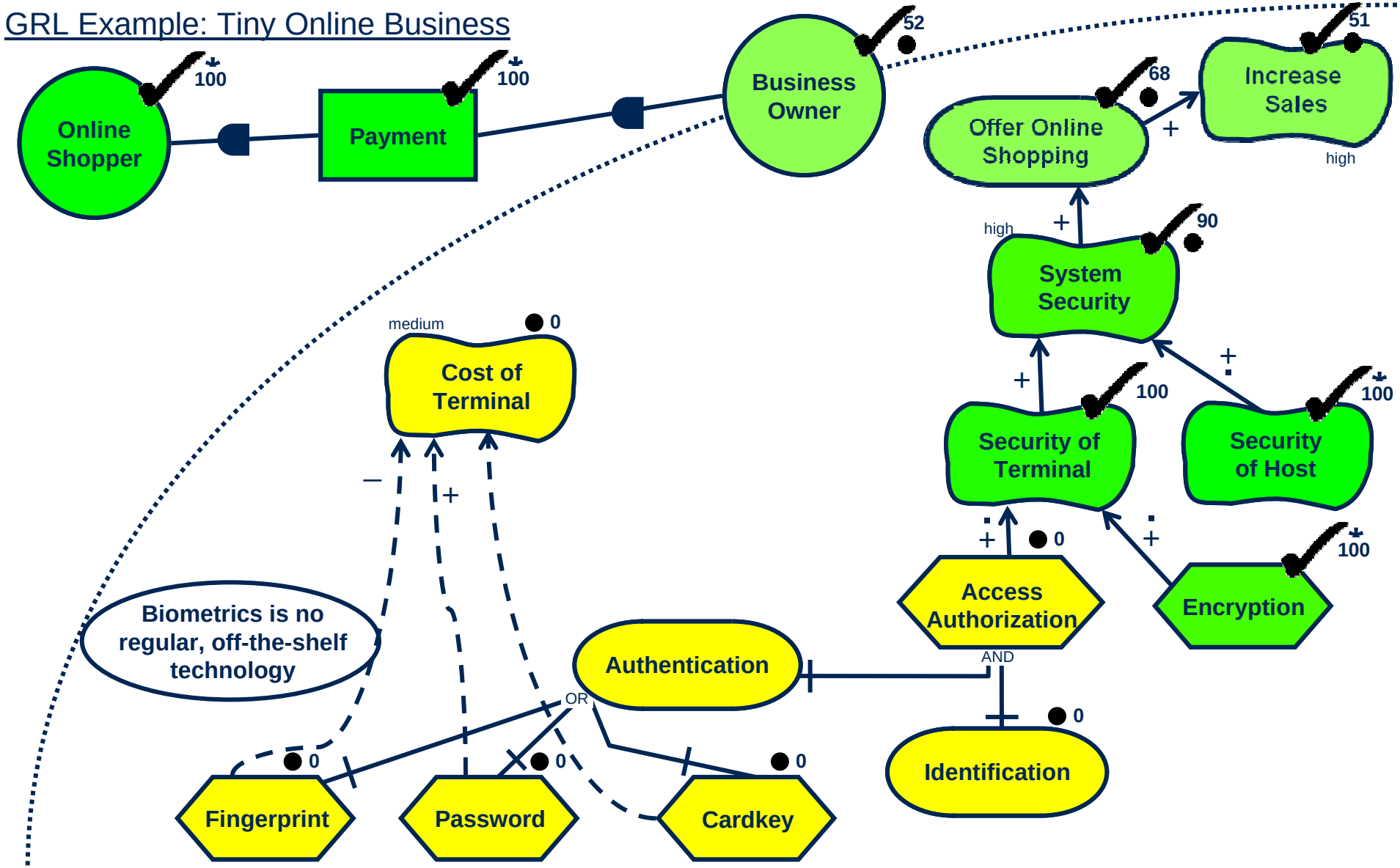
GRL – Strategy Execution (Strategy 1)



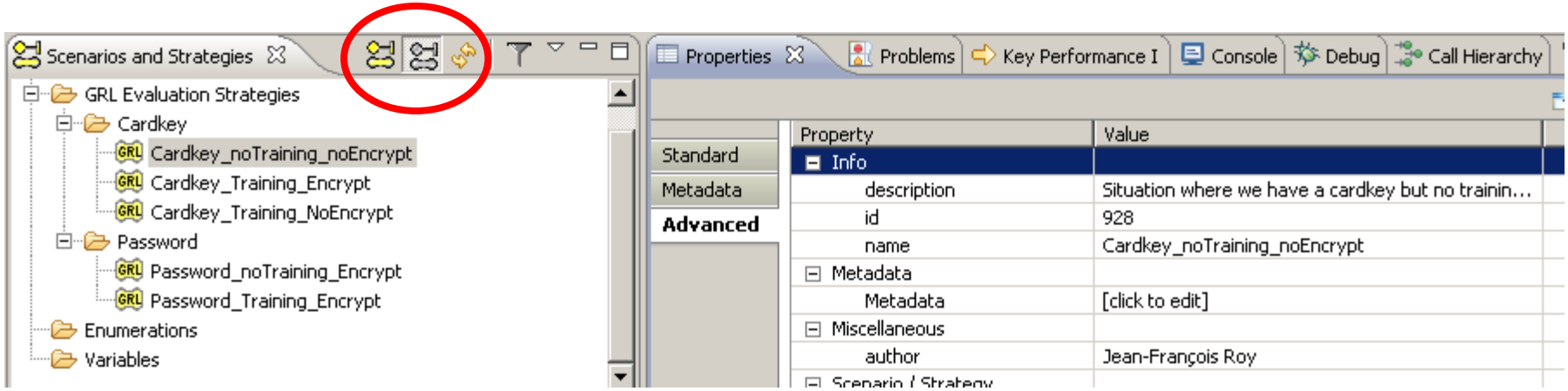
GRL – Strategy Execution (Strategy 2)

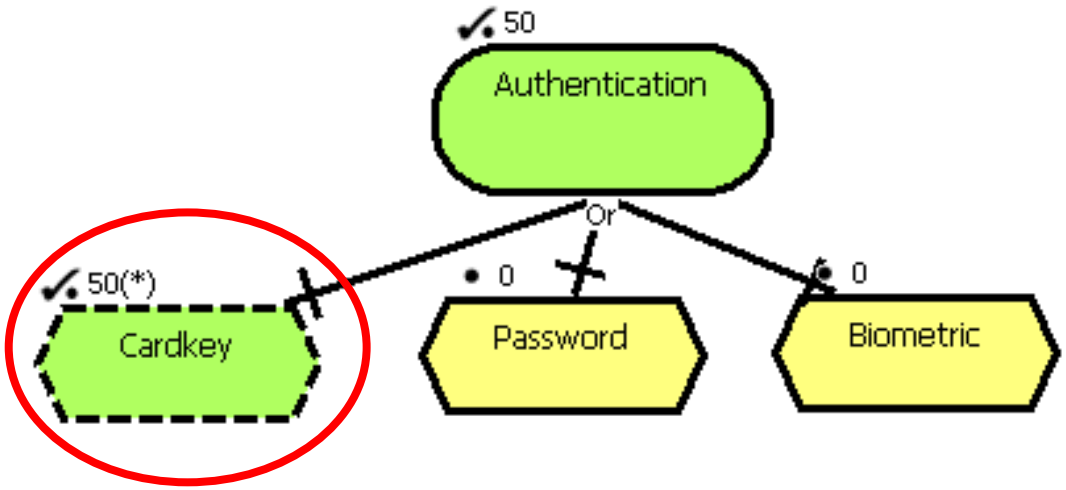


GRL – Strategy Execution (Strategy 3)



Strategies in jUCMNav

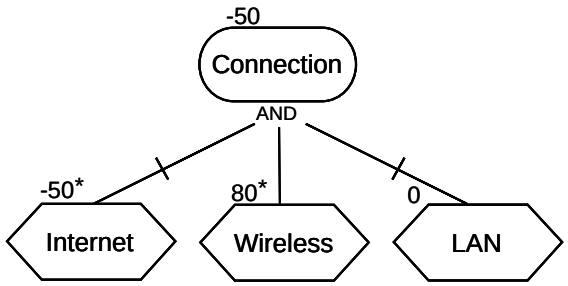




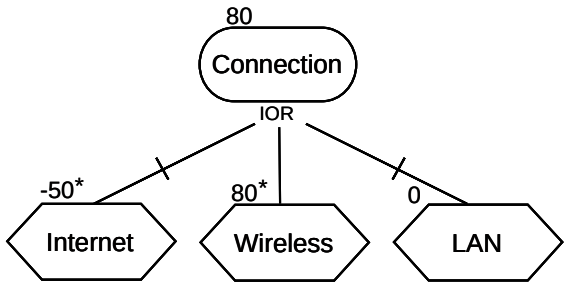
A star (*) indicates an initial value part of a given strategy (element also shown in dashed lines). All the others are evaluated through a propagation algorithm.

Quant. Alg. – Decompositions and Contributions

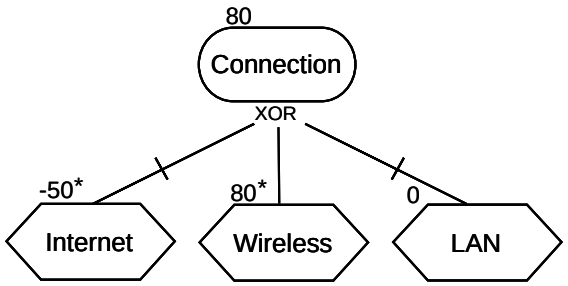
- Minimum for AND, maximum for OR



(a) AND decomposition

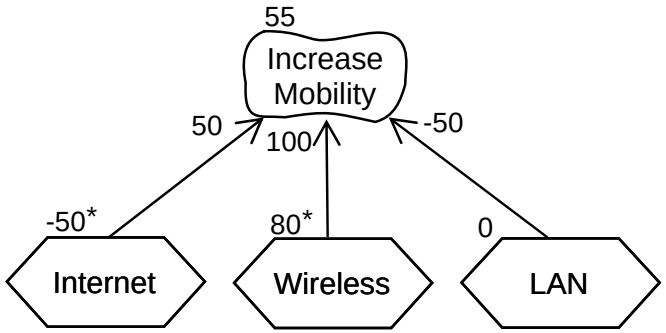


(b) IOR decomposition

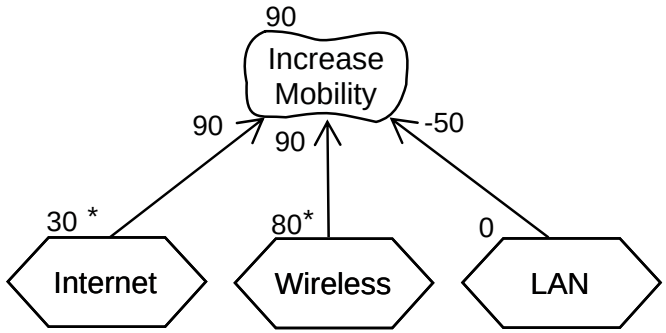


(c) XOR decomposition

- Contributions are additive but normalized and take a tolerance into account



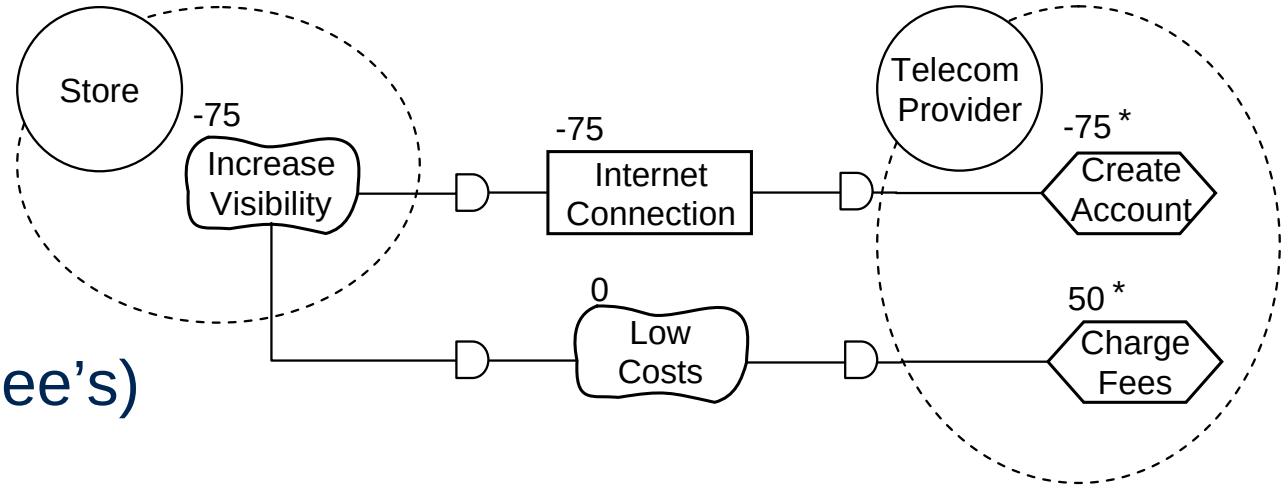
(a) Contributions



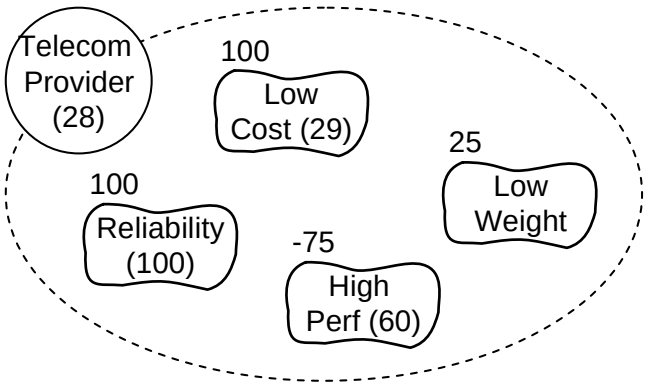
(b) Contributions with a tolerance of 10

Quantitative Algorithm – Dependencies and Actors

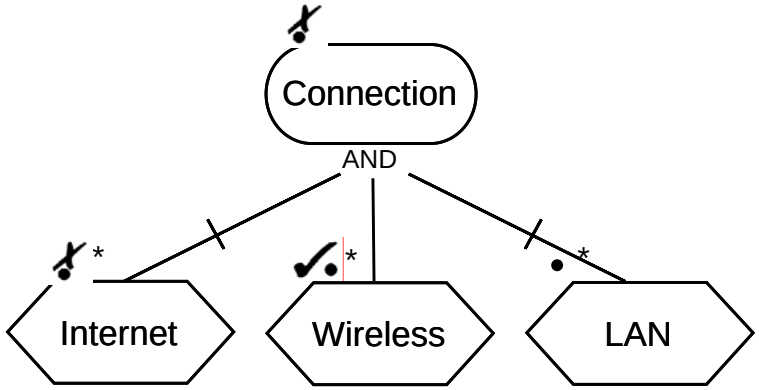
- Depender's satisfaction level is not more than the dependum's (and the dependee's)



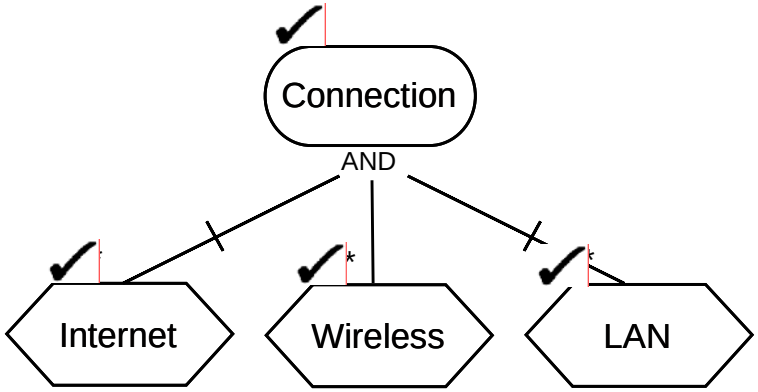
- Evaluations deal with negotiations between stakeholders
- Actor evaluations help analyzing and comparing the satisfaction levels of each actor based on the selected strategy
- Computed from importance attribute and satisfaction levels of intentional element references bound to actors



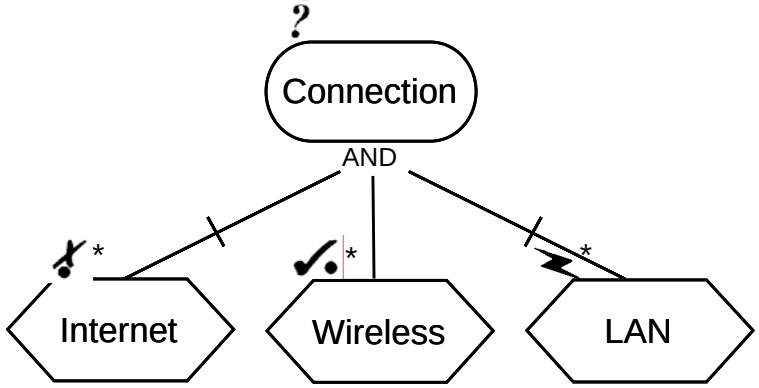
Qualitative Algorithm – AND Decomposition



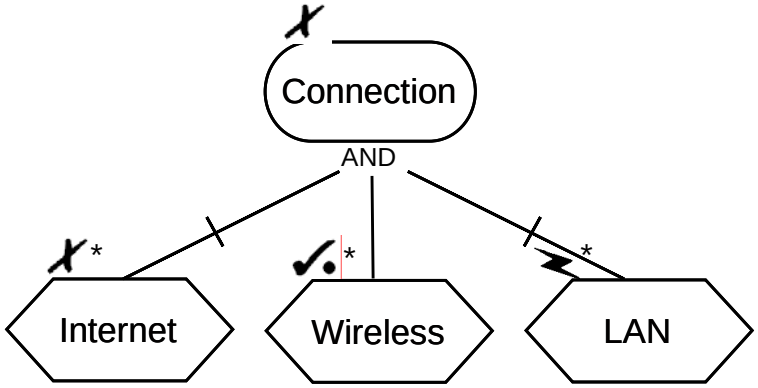
(a) Miminum is WeaklyDenied



(b) Miminum is Satisfied

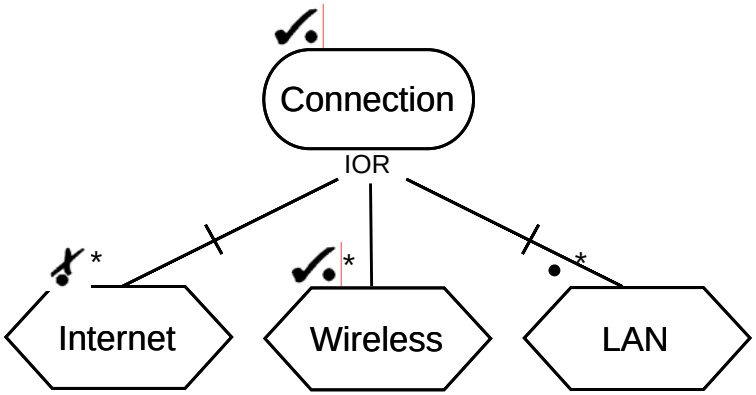


(c) Minimum is Conflict: Undecided is propagated

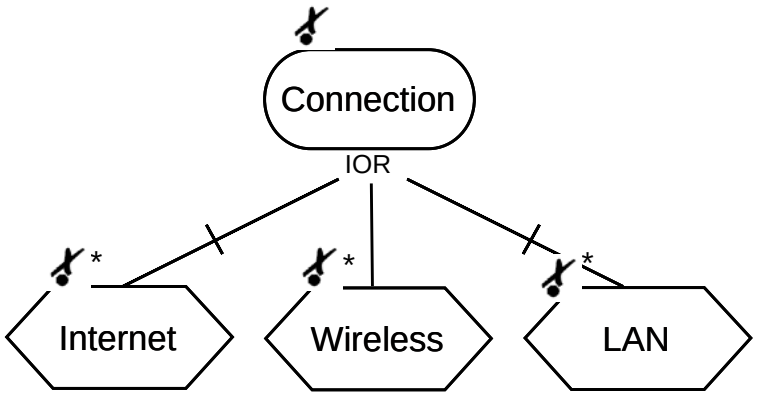


(d) Miminum is Denied, even if Conflict is present

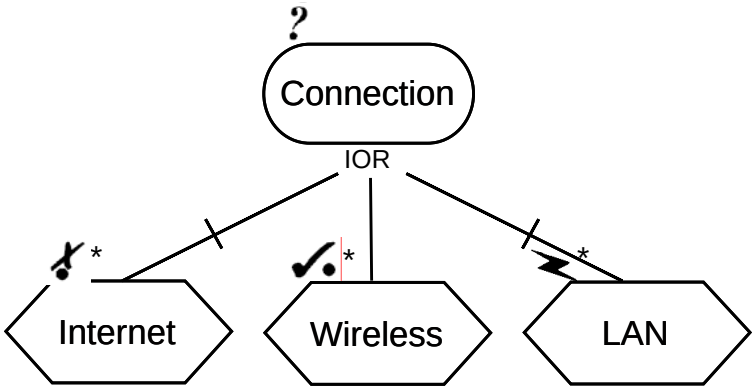
Qualitative Algorithm – OR Decomposition



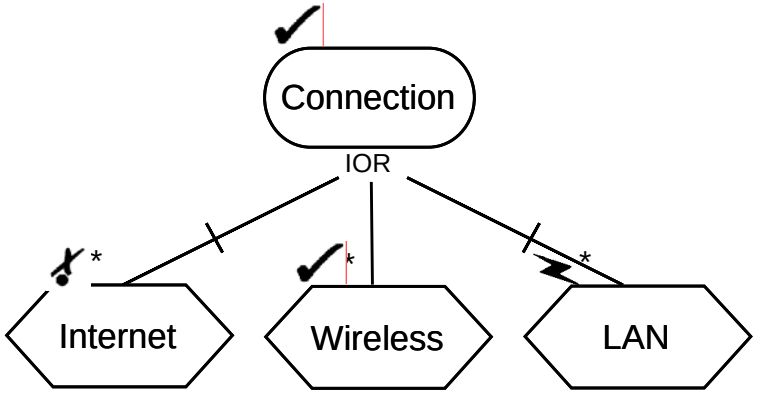
(a) Maximum is WeaklySatisfied



(b) Maximum is Denied

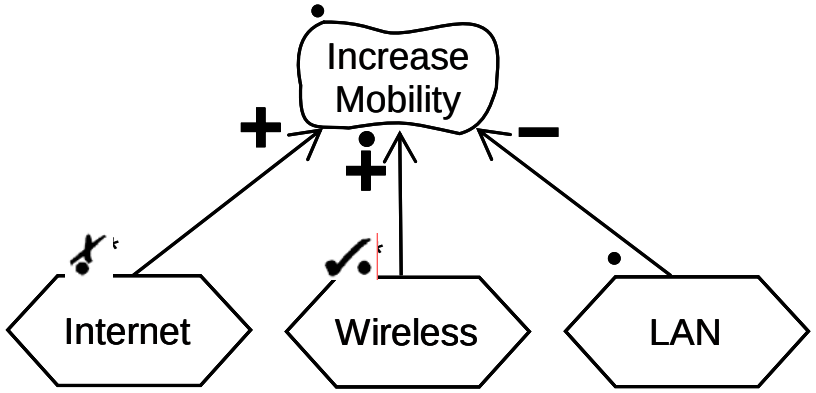


(c) Maximum is Conflict: Undecided is propagated

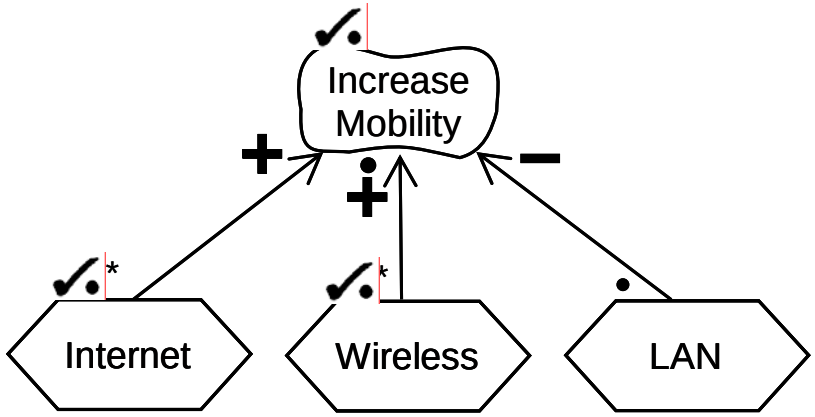


(d) Maximum is Satisfied, even if Conflict is present

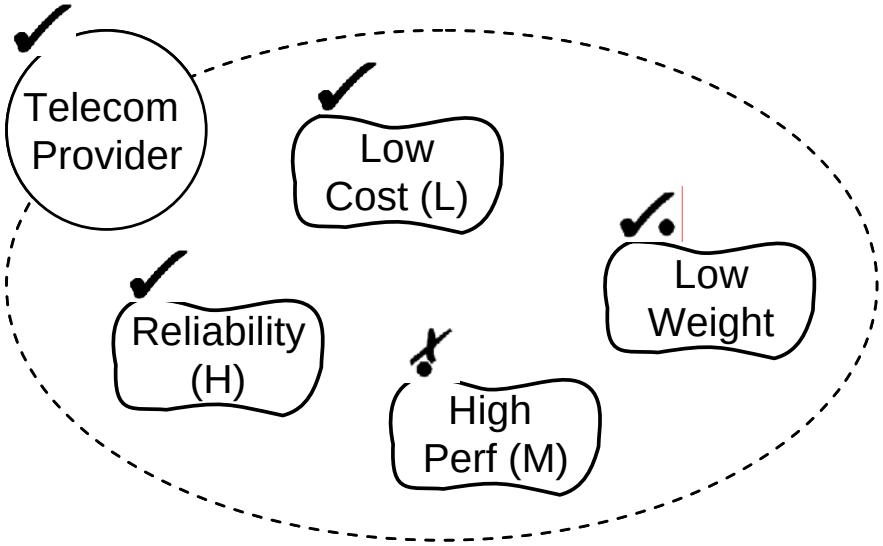
Qualitative Algorithm – Contributions and Actors



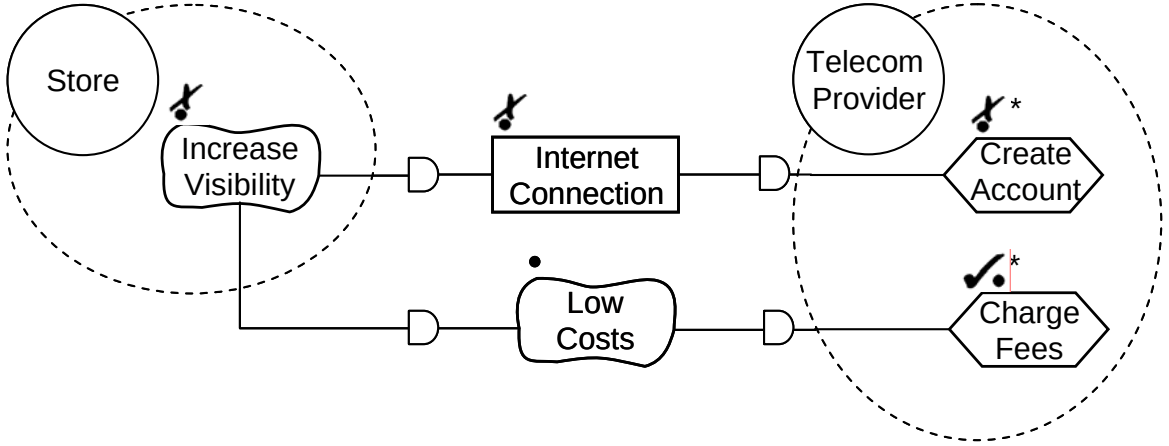
(a) Contributions: None is propagated



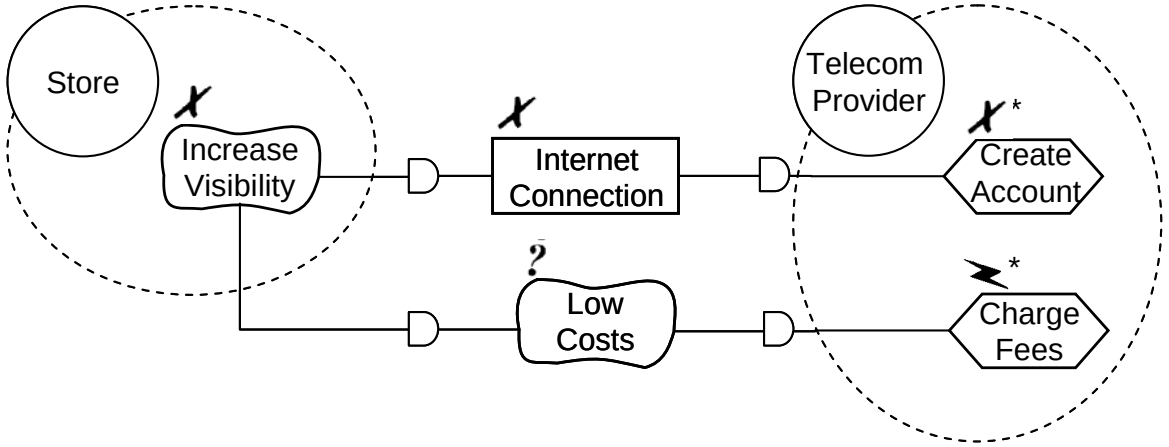
(b) Contributions: WeaklySatisfied is propagated



Qualitative Algorithm – Dependencies



(a) Minimum is WeaklyDenied

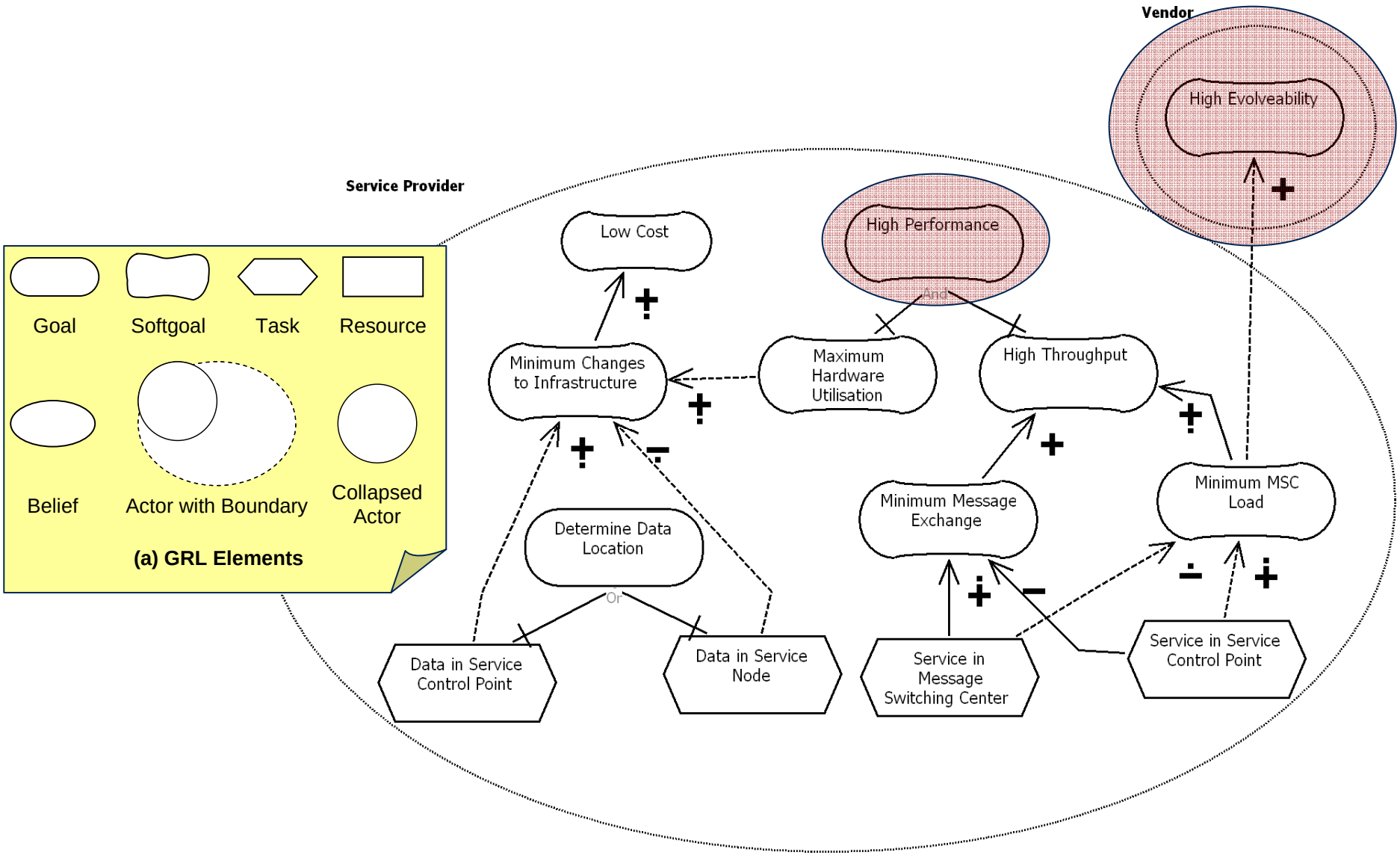


(b) Minimum is Denied, even if Conflict is present

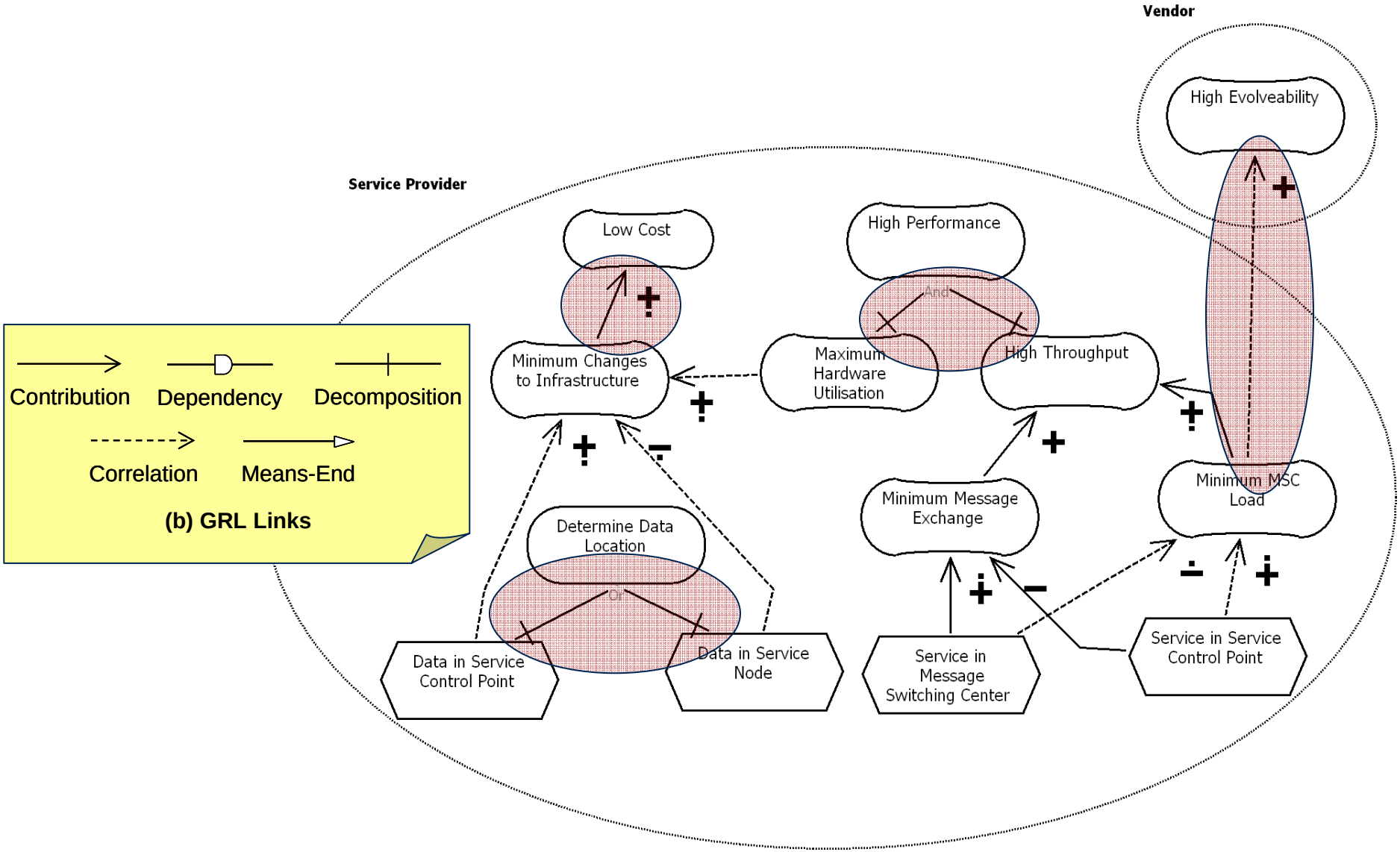
GRL Example I – Context

- New service for wireless network
  - Where to put the service logic?
  - Where to put the service data?

GRL Example I – Intentional Elements and Actors

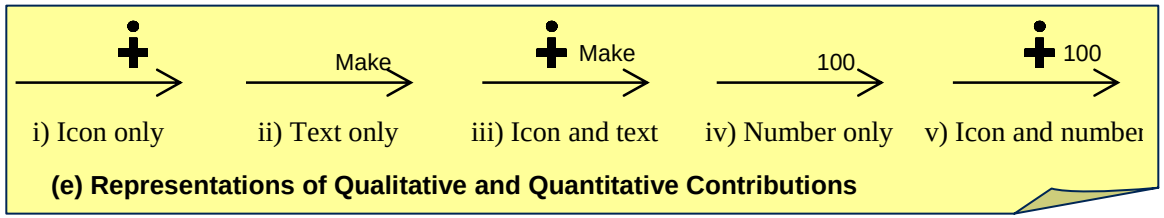


GRL Example I – Links

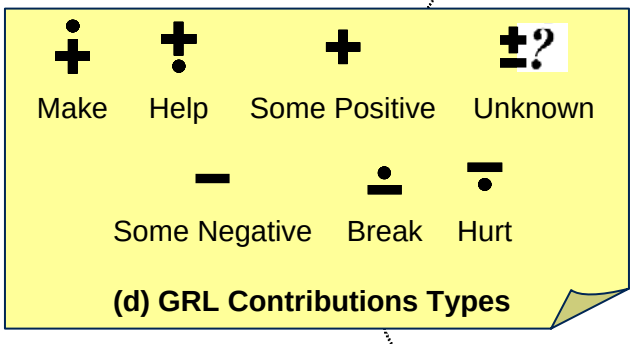


GRL Example I – Contribution Types

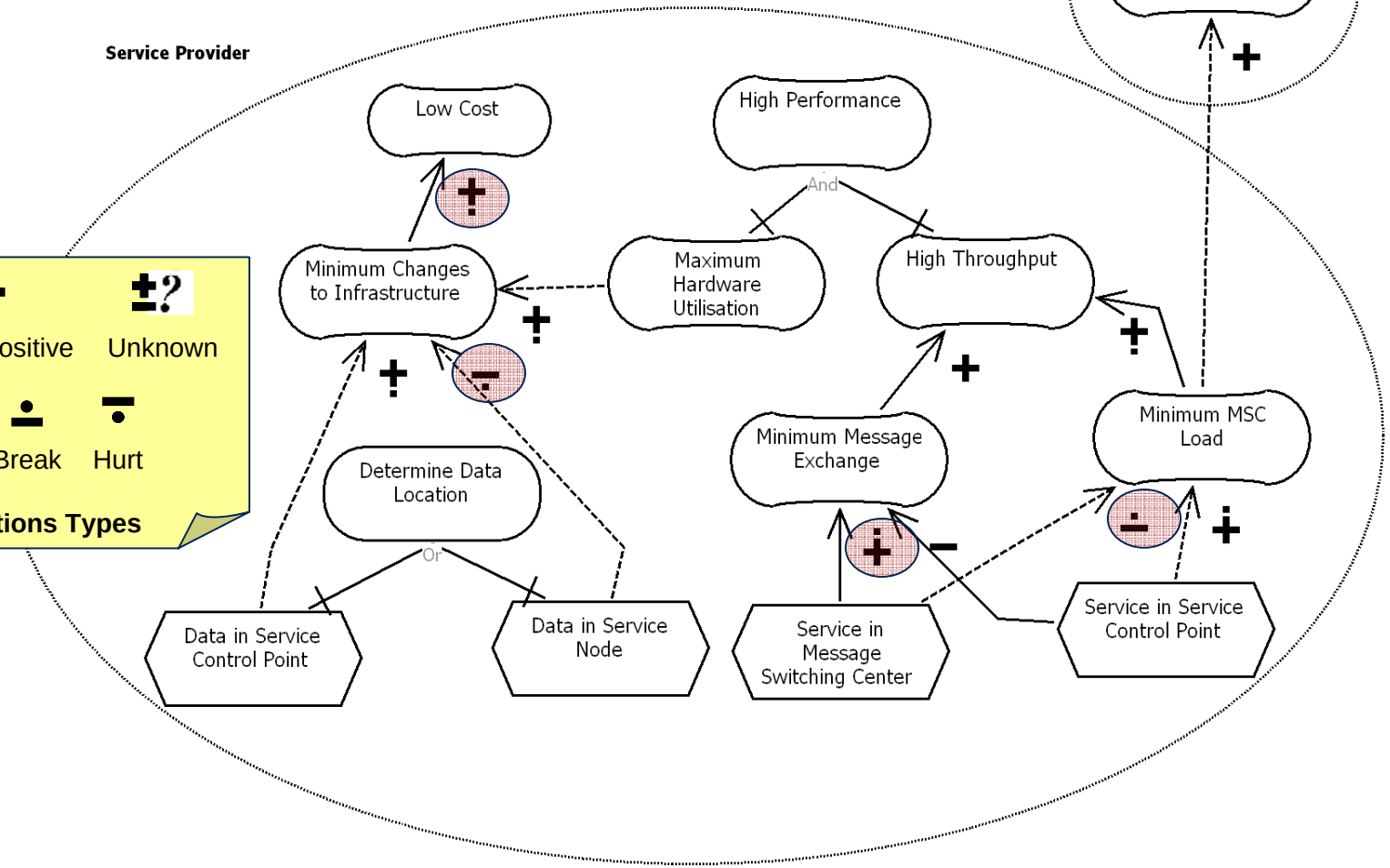
Qualitative *or* quantitative



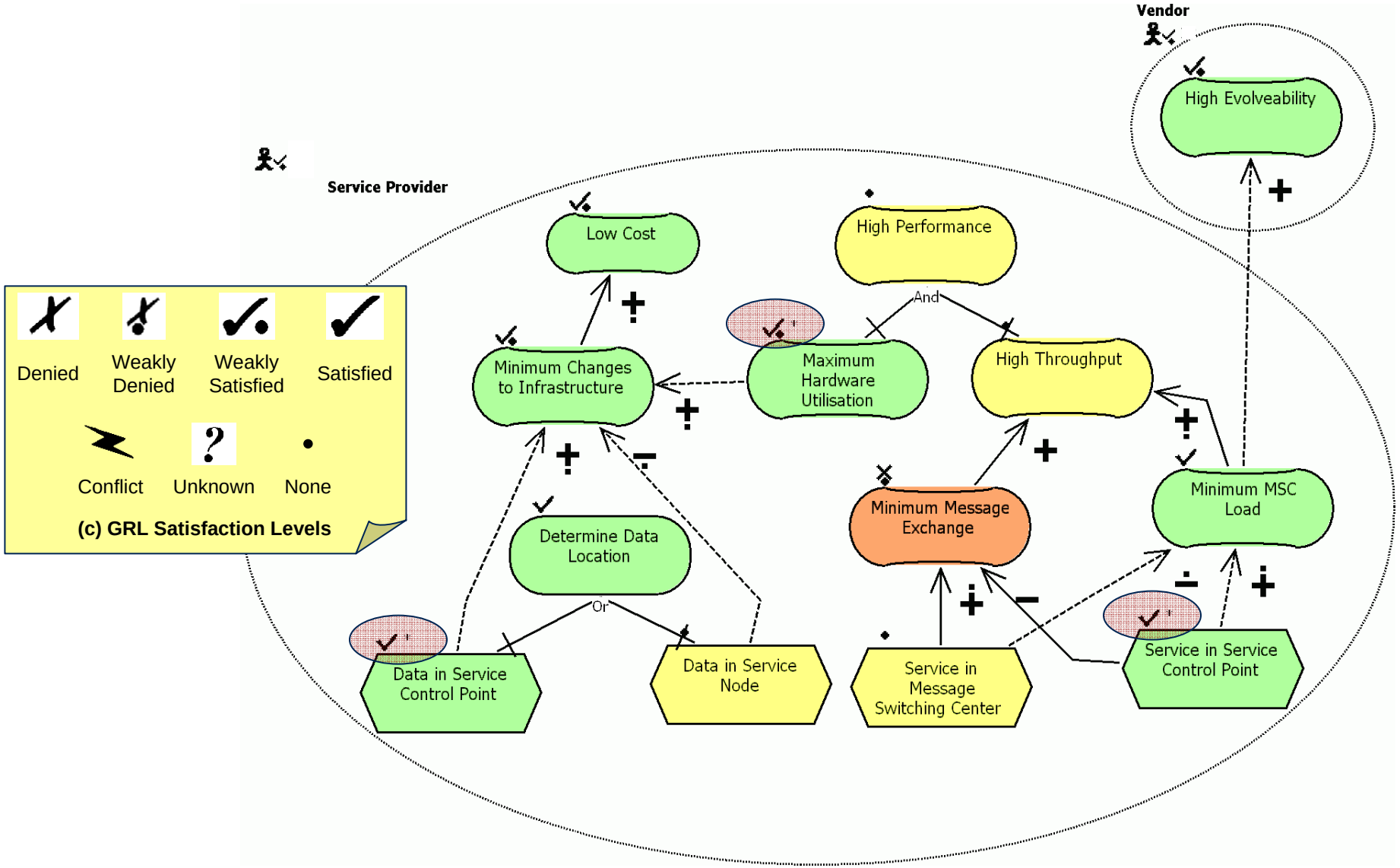
(e) Representations of Qualitative and Quantitative Contributions



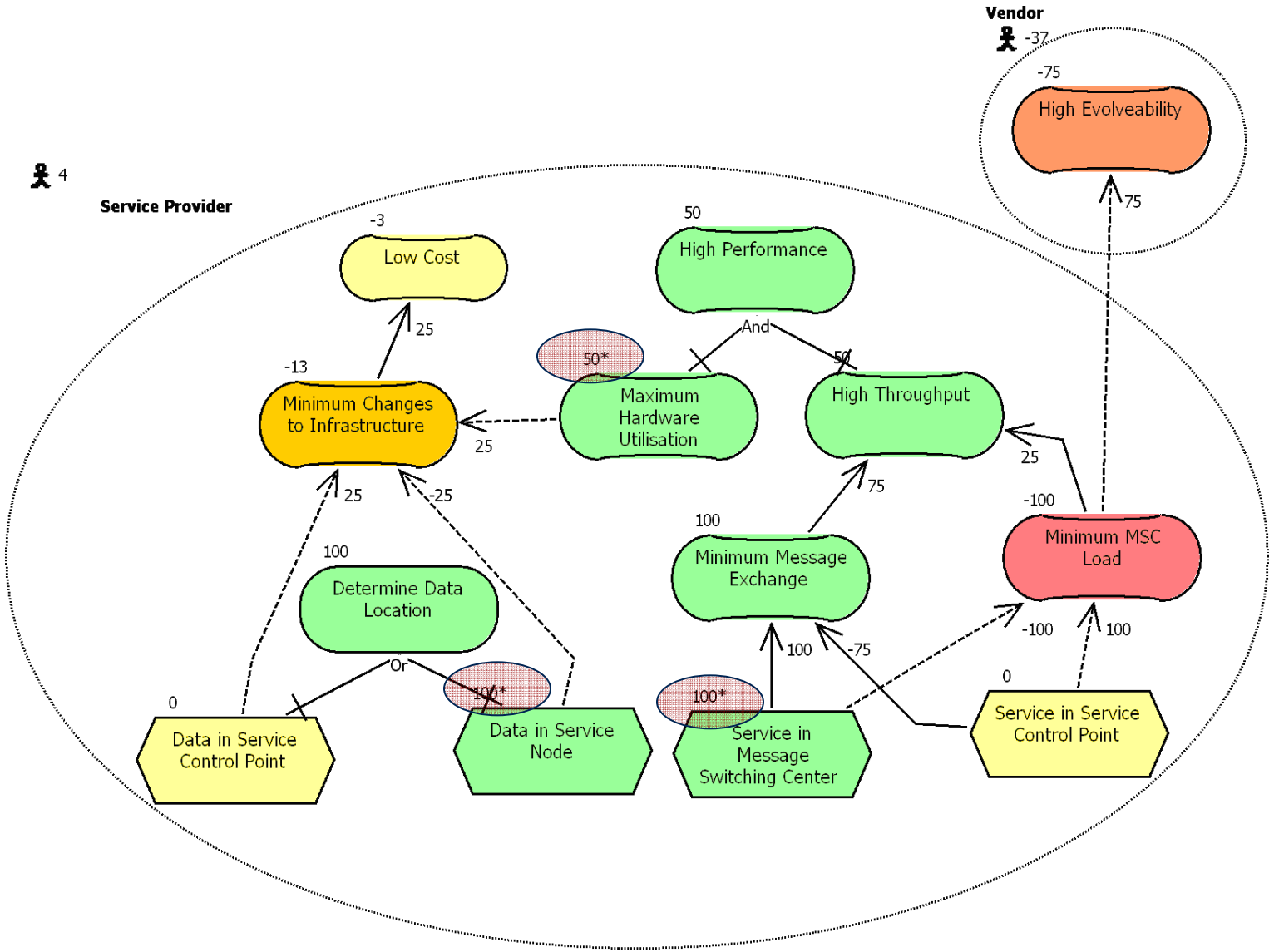
(d) GRL Contributions Types



GRL Example I – Qualitative Model Evaluation



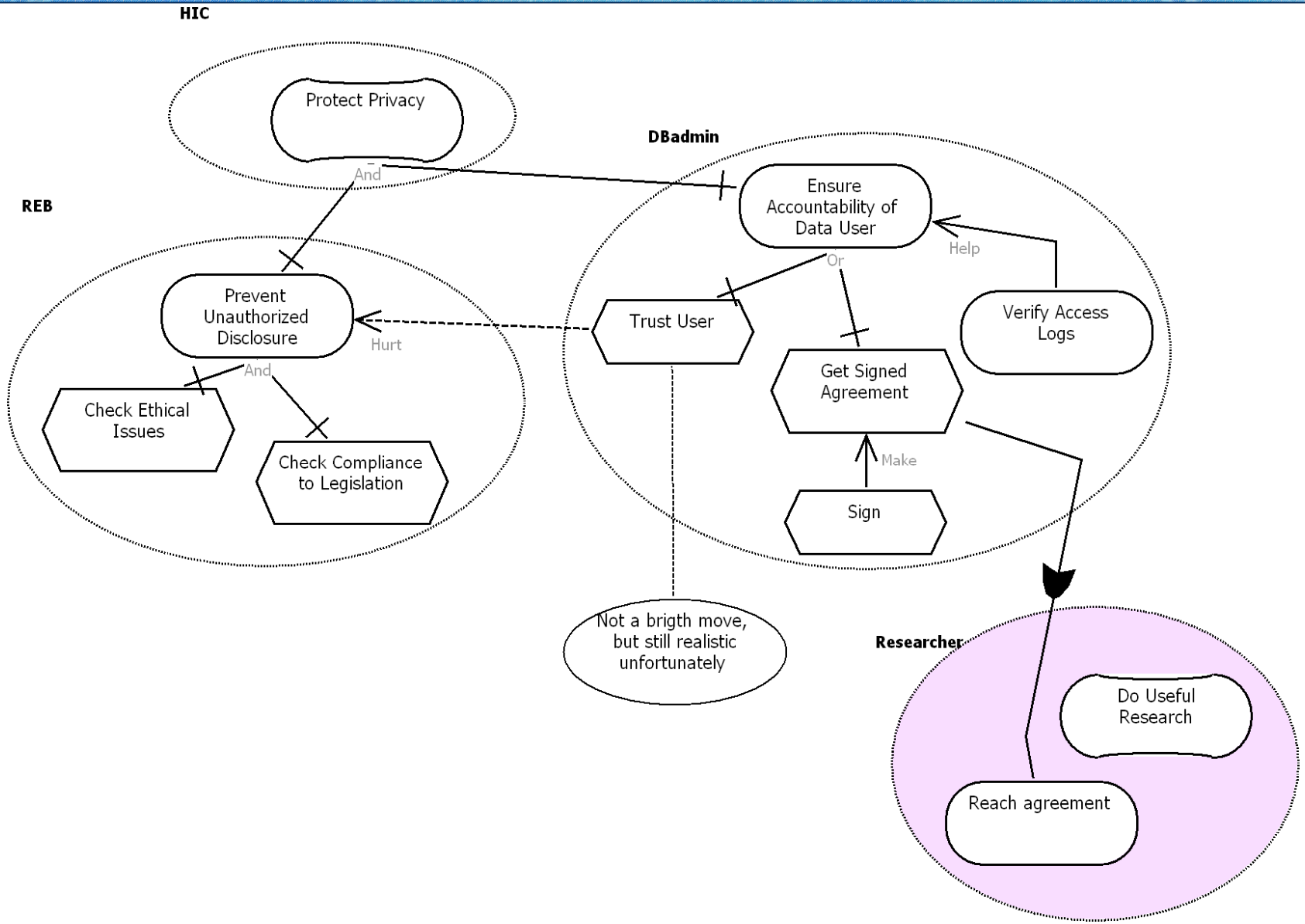
GRL Example I – Quantitative Model Evaluation



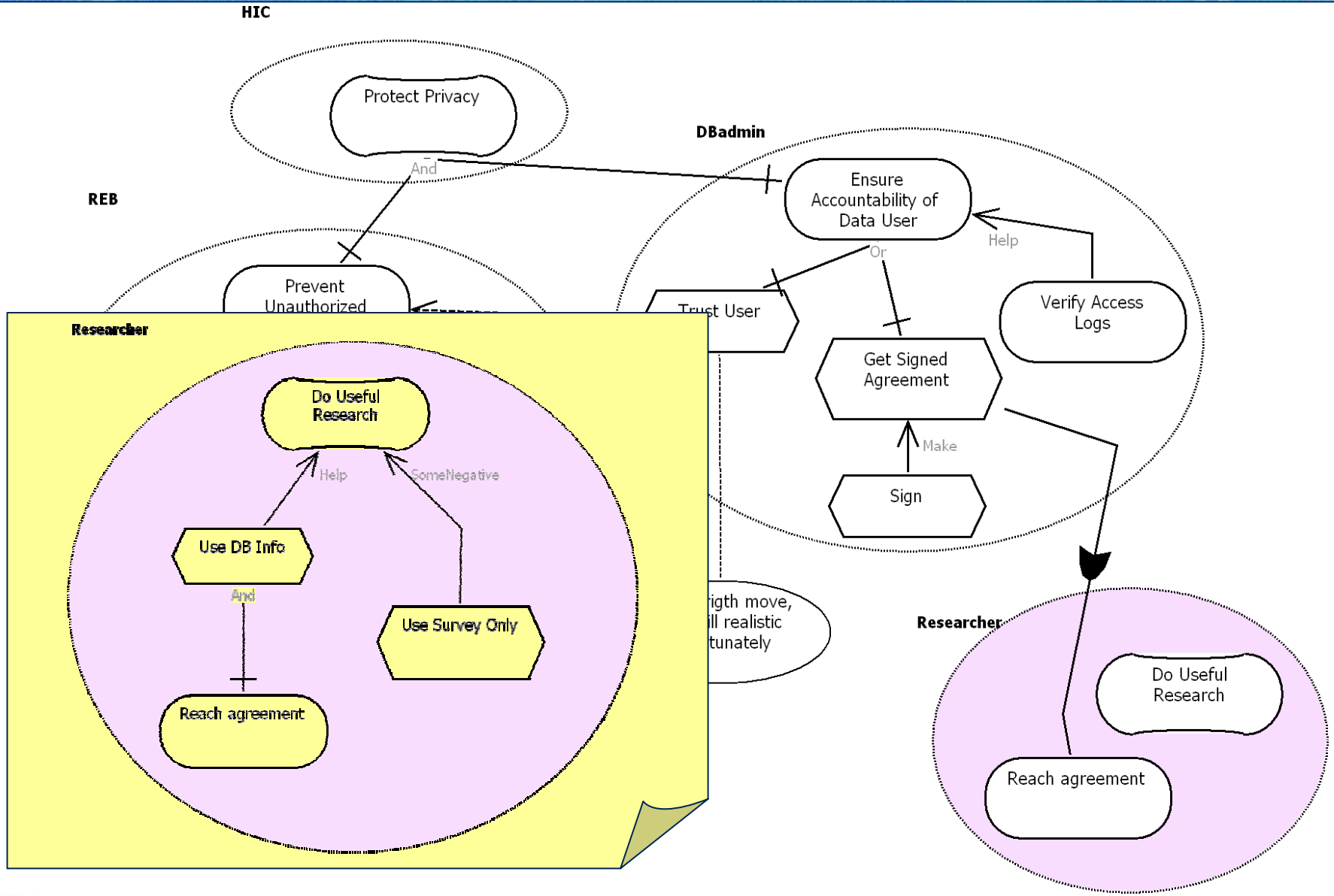
GRL Example II – Context

- GRL model that addresses privacy protection in a hospital environment
  - Researchers want access to patient data but the Health Information Custodian (HIC – i.e., the hospital) needs to protect patient privacy, as required by law (PHIPA in Ontario).
  - The process of accessing databases must ensure privacy. As required by law, a Research Ethics Board (REB) is usually involved in assessing privacy risks for the research protocol proposed by a researcher.
  - DB administrators also want to ensure that DB users are accountable for their acts.

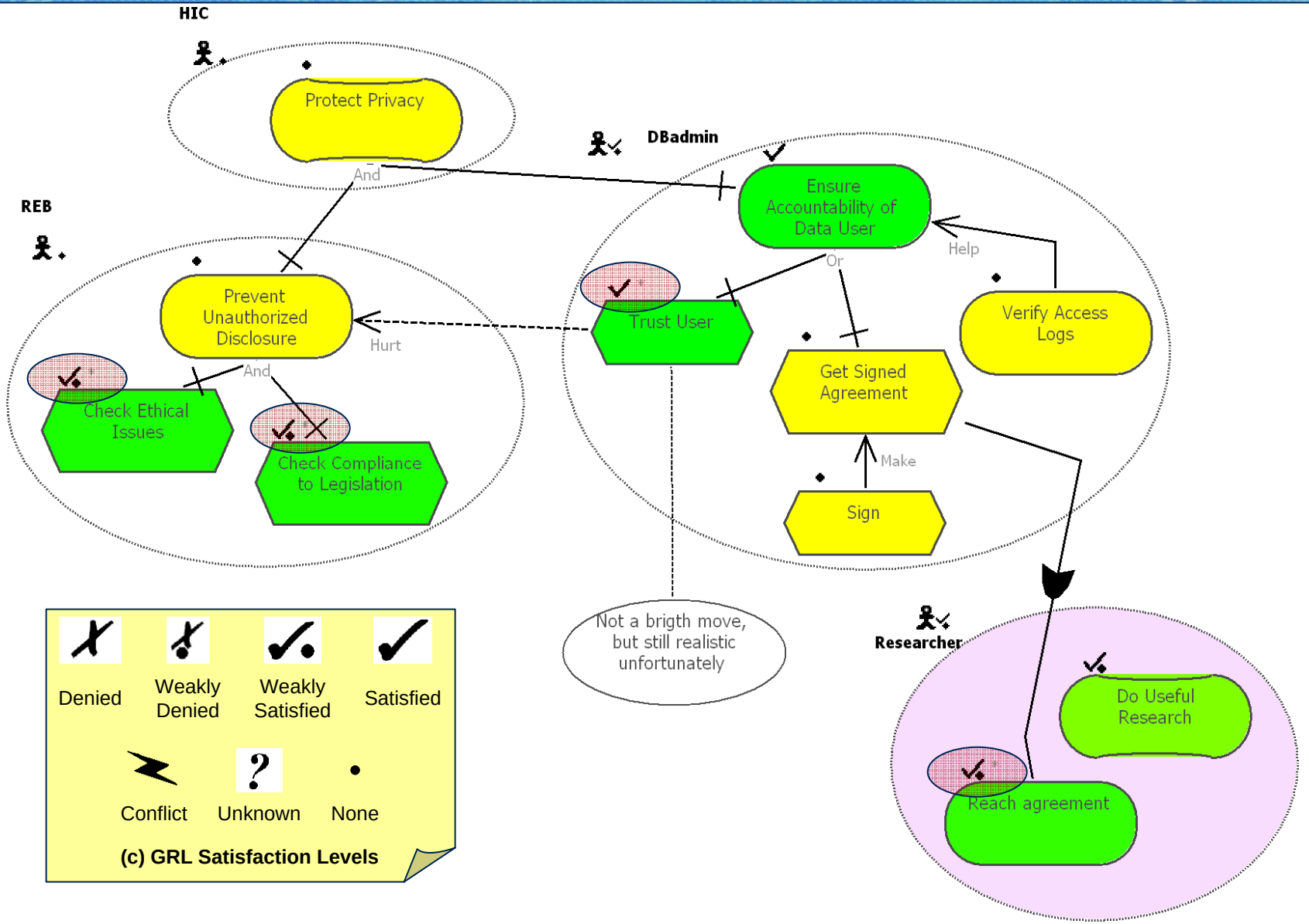
GRL Example II – GRL Model



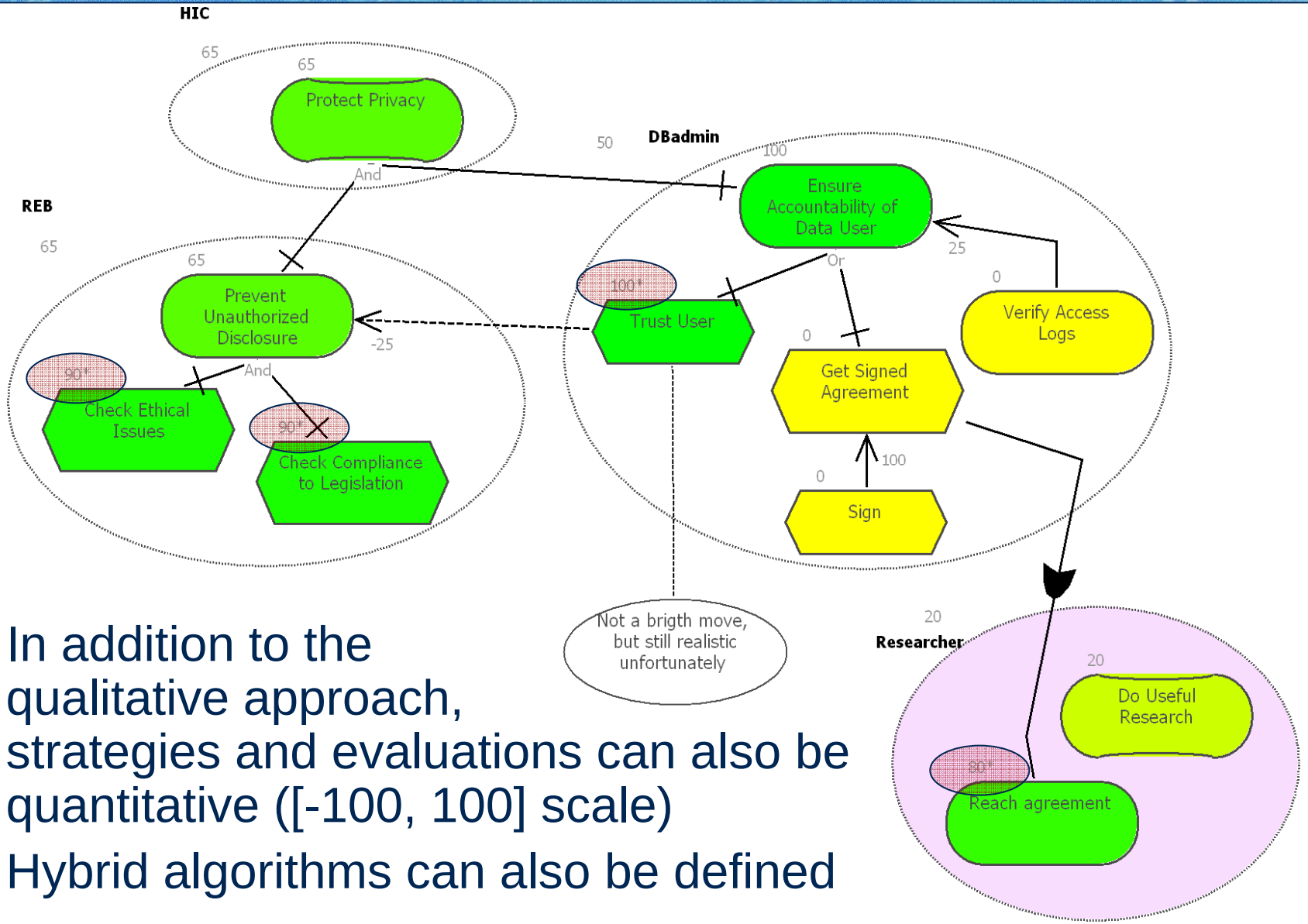
GRL Example II – One GRL Model, Many Diagrams



GRL Example II – Qualitative Model Evaluation



GRL Example II – Quantitative Model Evaluation



- In addition to the qualitative approach, strategies and evaluations can also be quantitative ([-100, 100] scale)
- Hybrid algorithms can also be defined

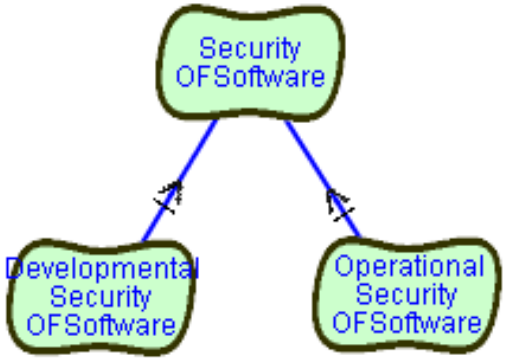
GRL Example III – Context

- Information system for a Financial Institution
  - Provides support for point of sale systems for financial transactions
  - Remote input at Retailers allows customers to make payments
  - Need to address security when producing, deploying, and updating the financial software at the financial institution and at the Retailers site
    - Software itself must be secured (in source and object form)
    - Access (e.g. for update) also need to be secured
  - Possible tradeoffs with
    - Ease of use
    - Performance
    - Cost

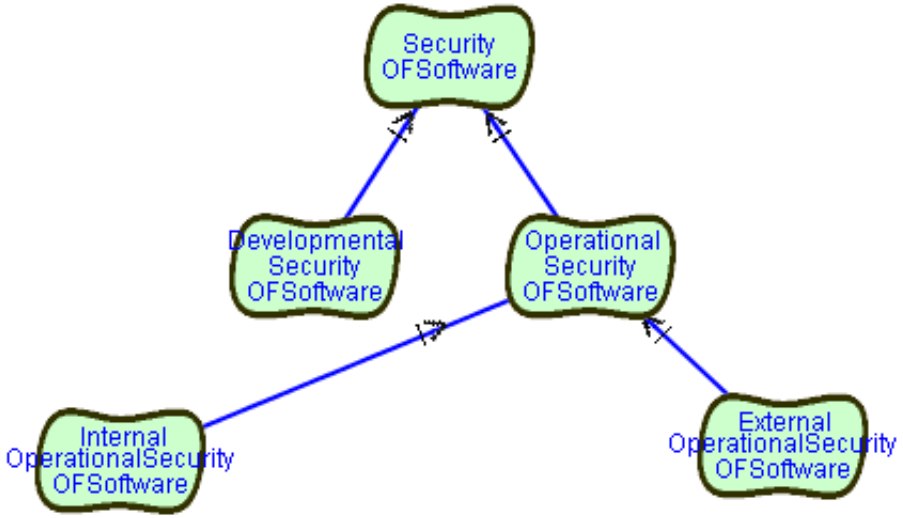
Source: http://www.cs.toronto.edu/km/GRL/

GRL Example III – First Goal Refinements

- Security includes
  - Security during operation (e.g. updating, regular operation…)
  - Security during software development

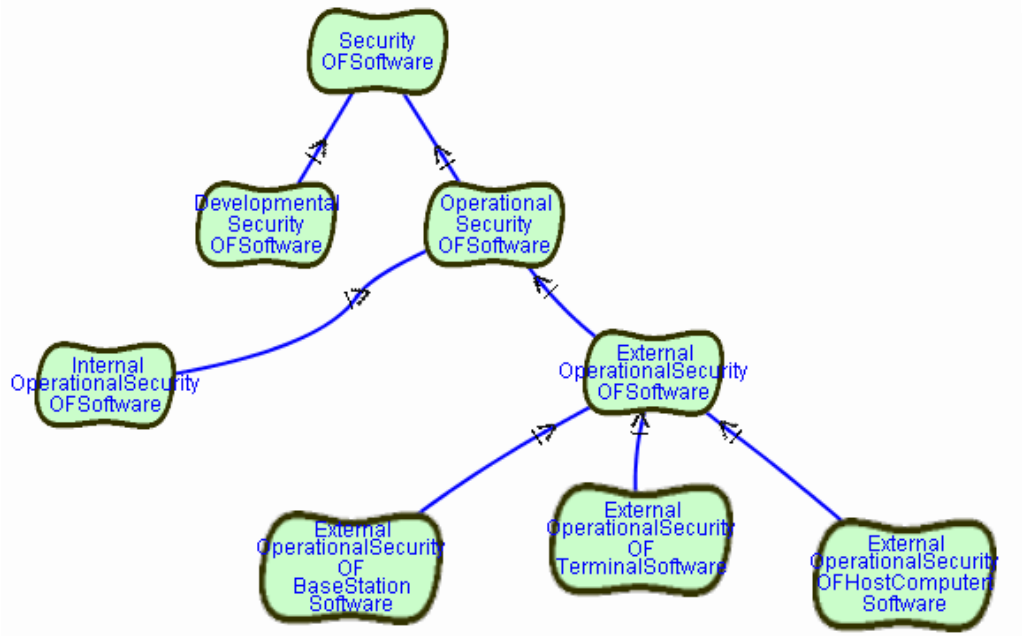


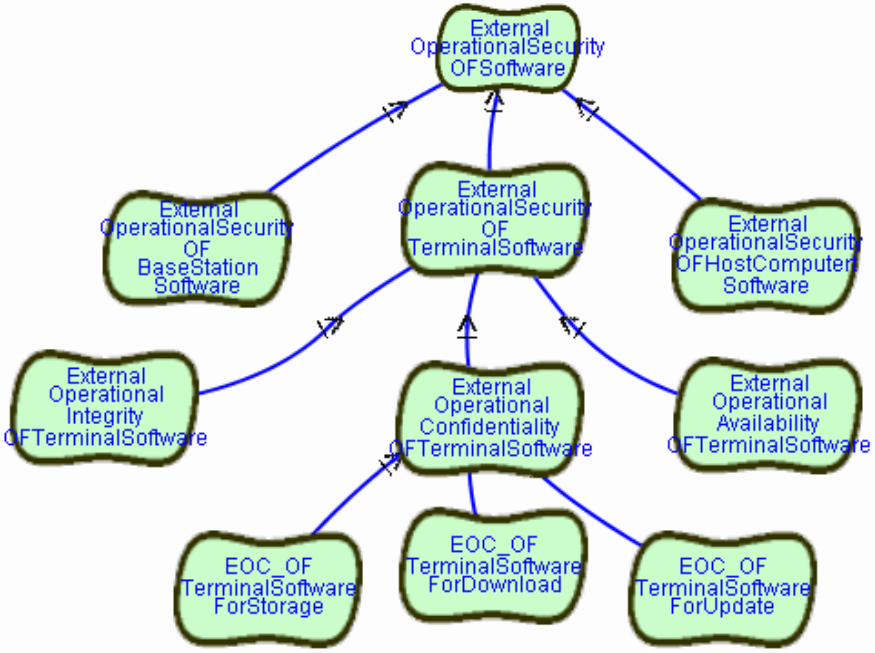
- Operational Security includes
  - Financial institution
  - Retailer sites



GRL Example III – More Goal Refinements

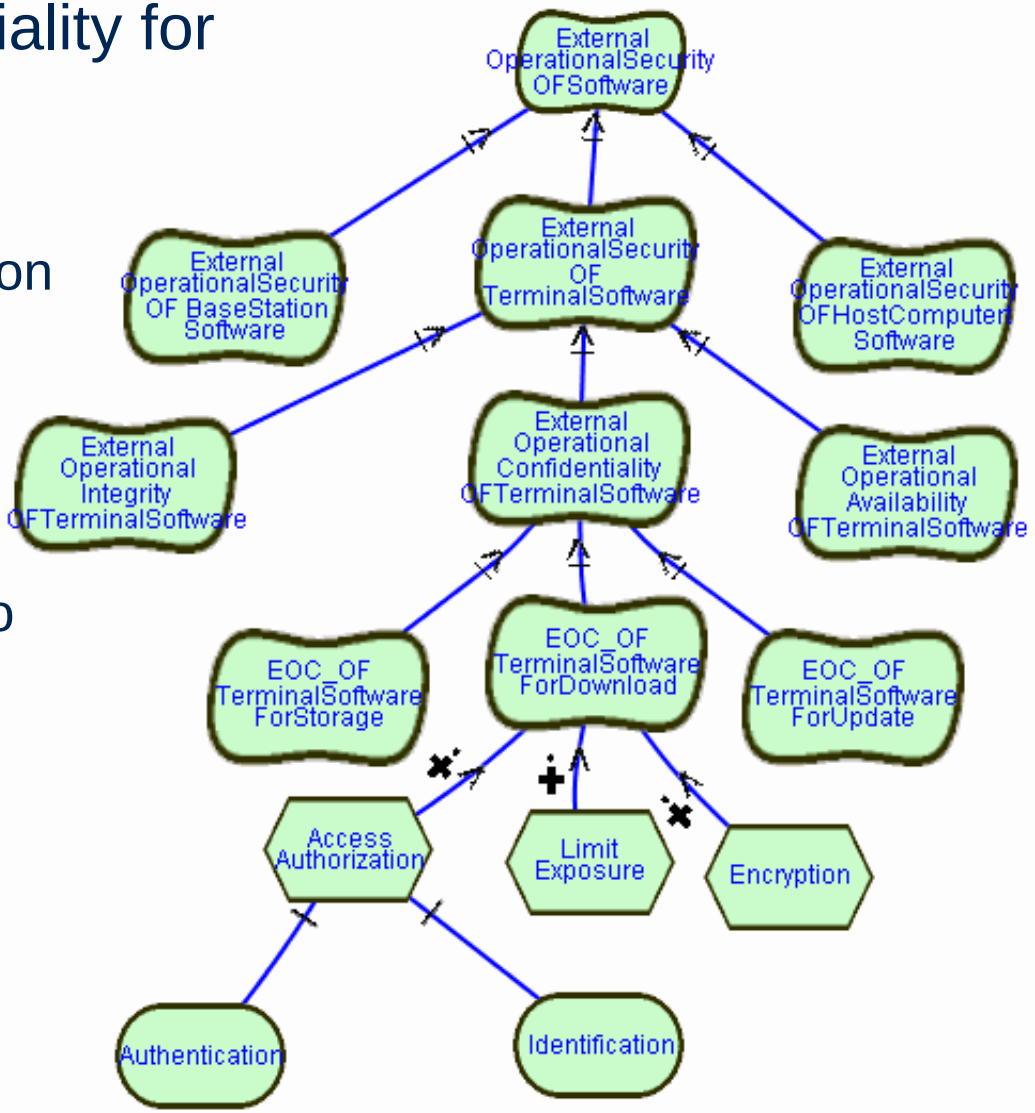
- System includes
  - Base Station connected to a set of Terminals
  - Host Computer at Financial Institution
- Refinement of terminal security into
  - Confidentiality, integrity, availability
- Each operation (update, download, storage) needs to be secured





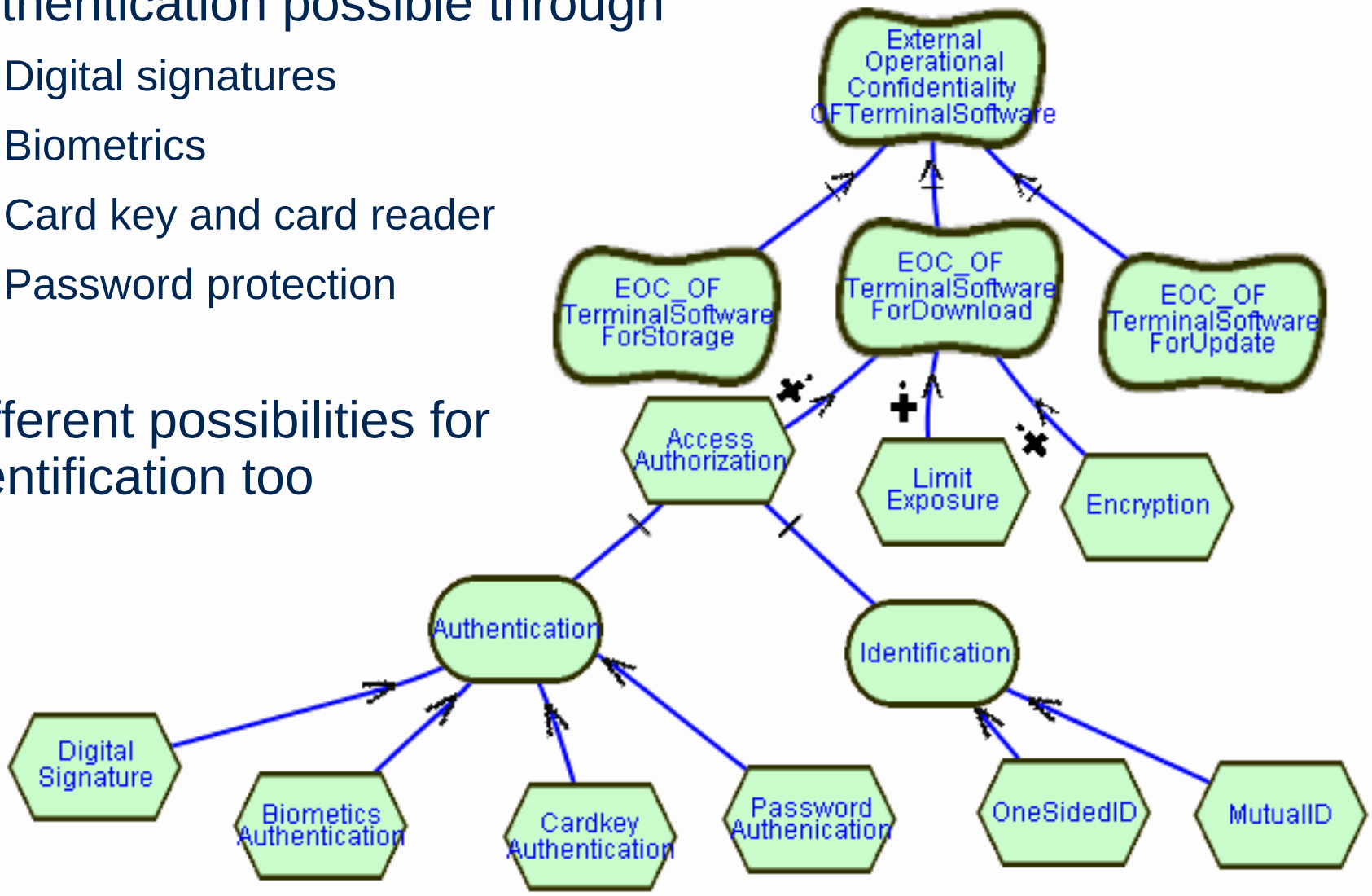
GRL Example III – Thinking About Solutions

- How to provide confidentiality for download?
- Three possibilities
  - Provide access authorisation
    - Authentication
    - Identification
  - Provide encryption
  - Provide limited exposure to accessing the software



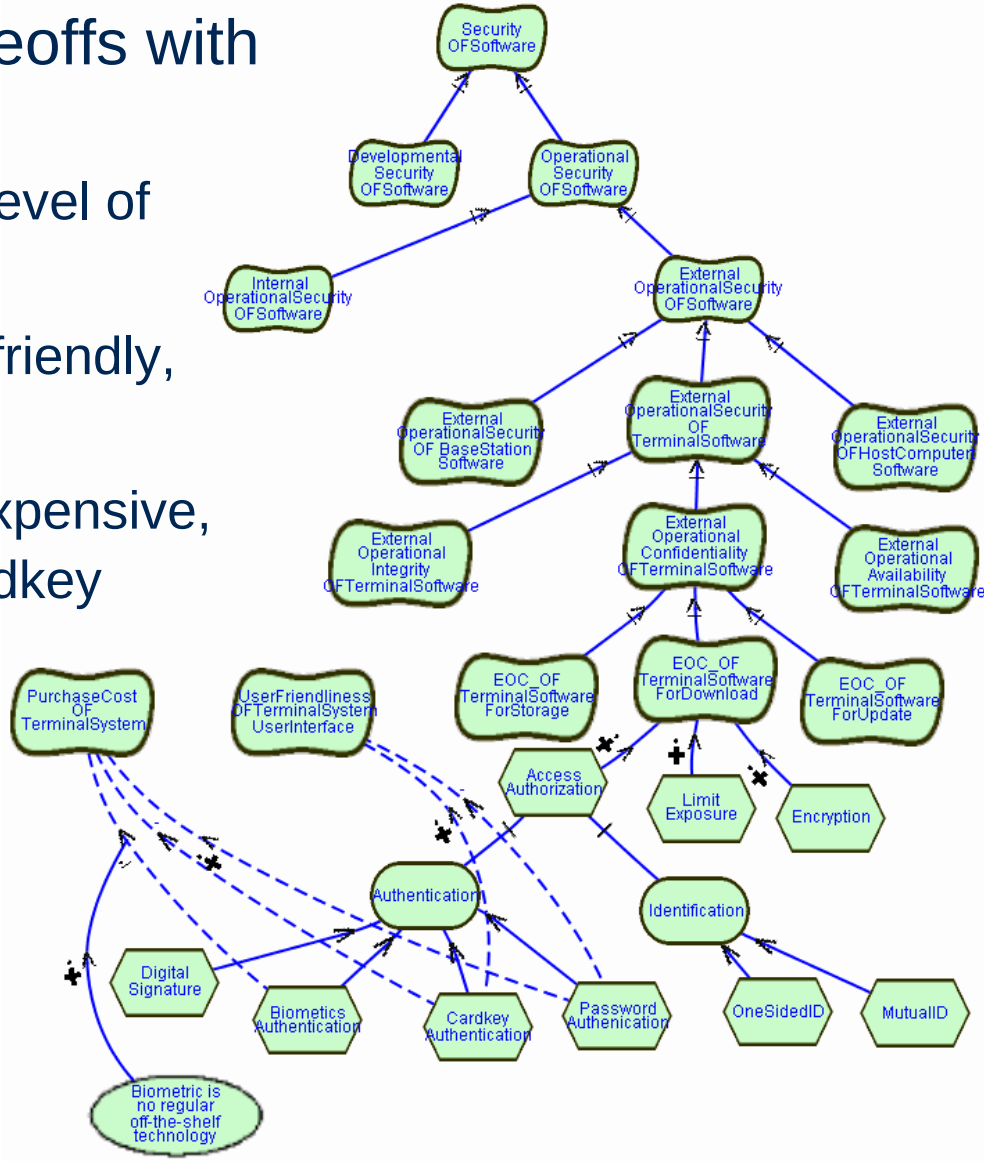
GRL Example III – More Solutions

- Authentication possible through
  - Digital signatures
  - Biometrics
  - Card key and card reader
  - Password protection
- Different possibilities for identification too

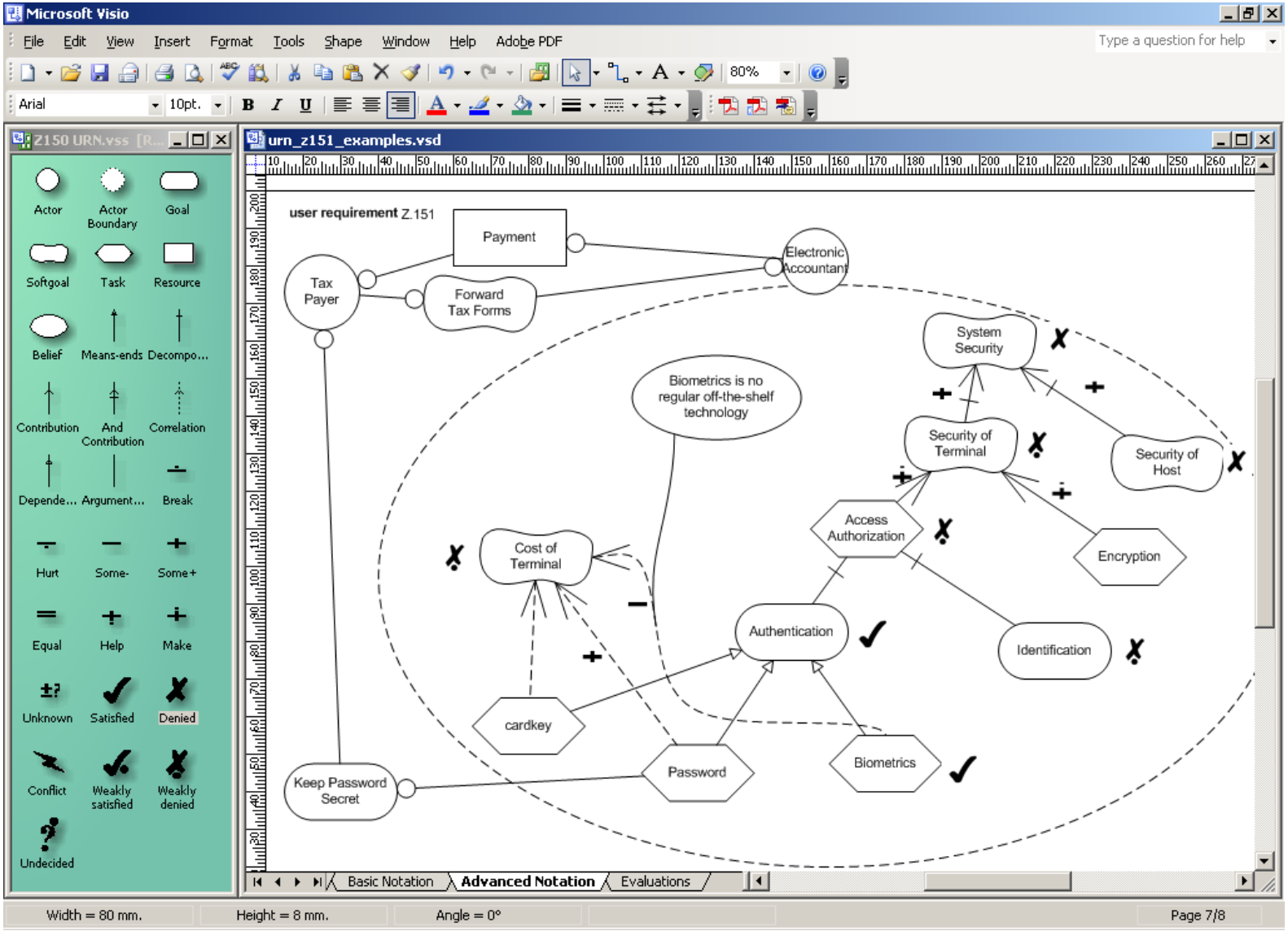


GRL Example III – Side Effects

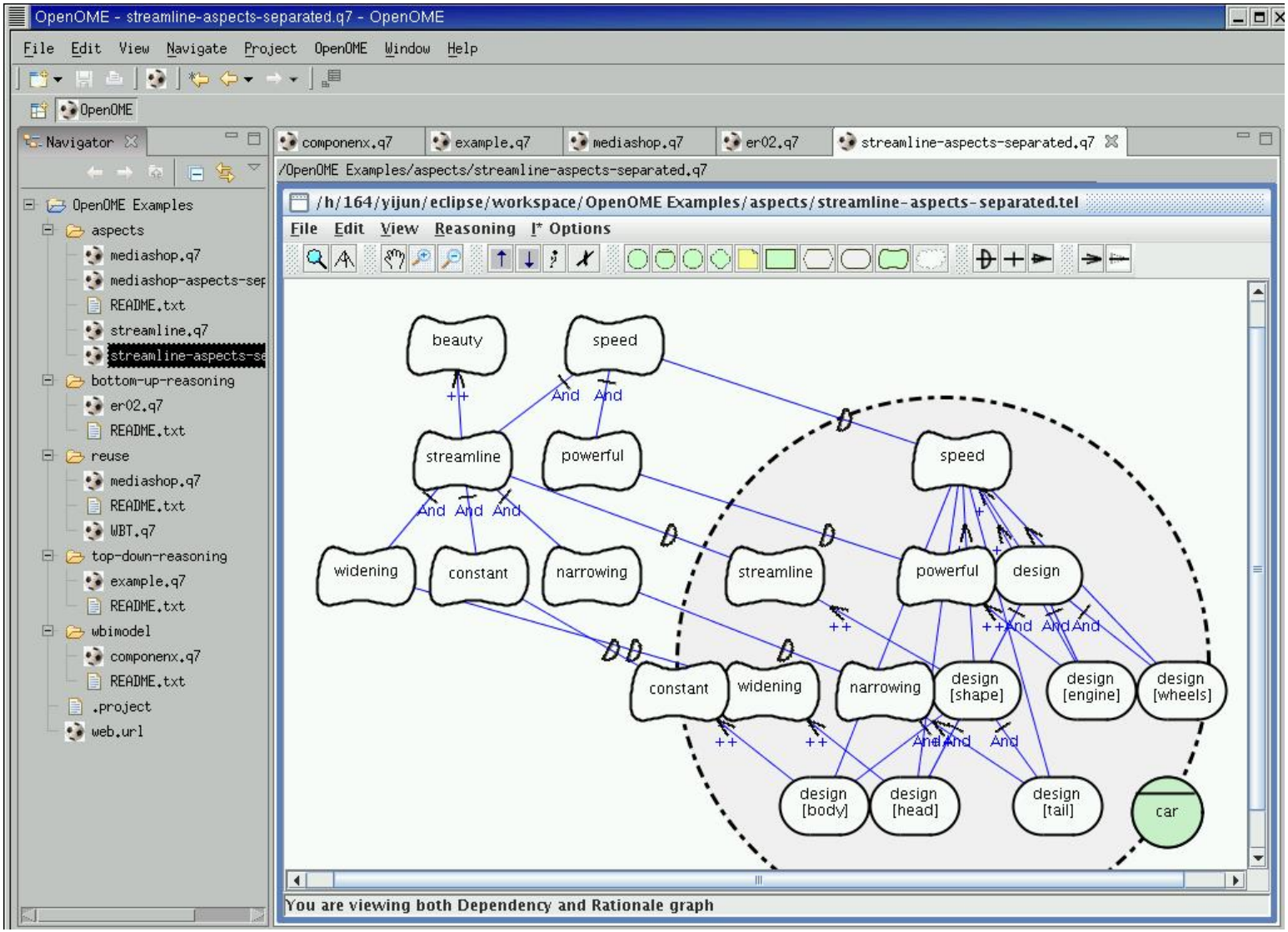
- Alternatives have set of tradeoffs with other requirements
  - Biometrics will provide a high level of security but will be expensive
  - Cardkey less expensive, user friendly, but equipment needed too
  - Password protection is least expensive, but not as user friendly as Cardkey
- Can be model using correlation links to other nodes



Tool Support – SanDriLa (Plug-in for Visio)



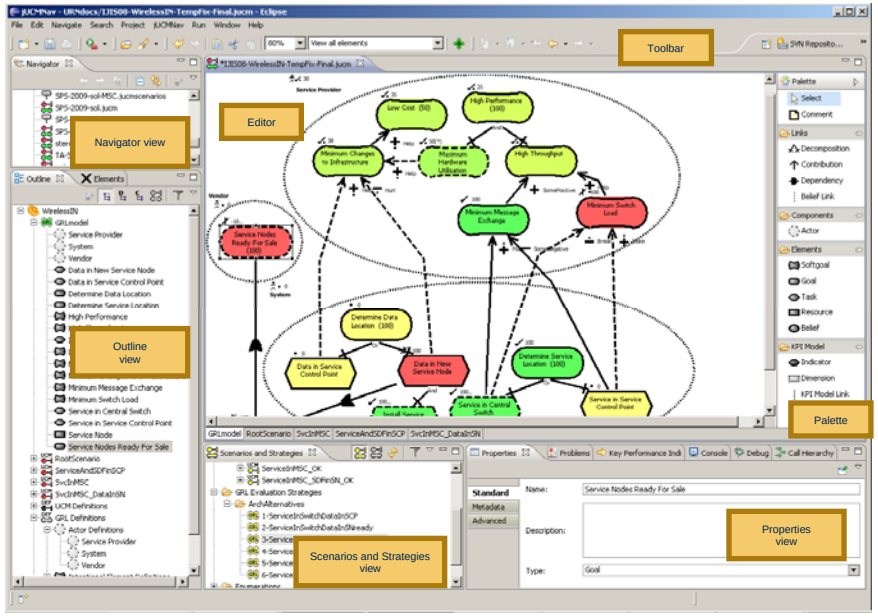
Tool – OpenOME, with Eclipse Plug-in



http://www.cs.toronto.edu/km/openome/

jUCMNav (Eclipse Plug-in)

- Features for GRL
  - 4 GRL evaluation algorithms, with color highlight
  - One model, multiple diagrams
  - References to actors and intentional elements
    - Drag&drop from outline or via properties
  - Auto-layout
  - Catalogues
    - For exporting/importing/reusing common models
  - Export graphics (.bmp, .gif, .jpg)
  - Export strategy evaluations (.csv)
  - URN links (for integration with UCMs)
  - Export to DOORS

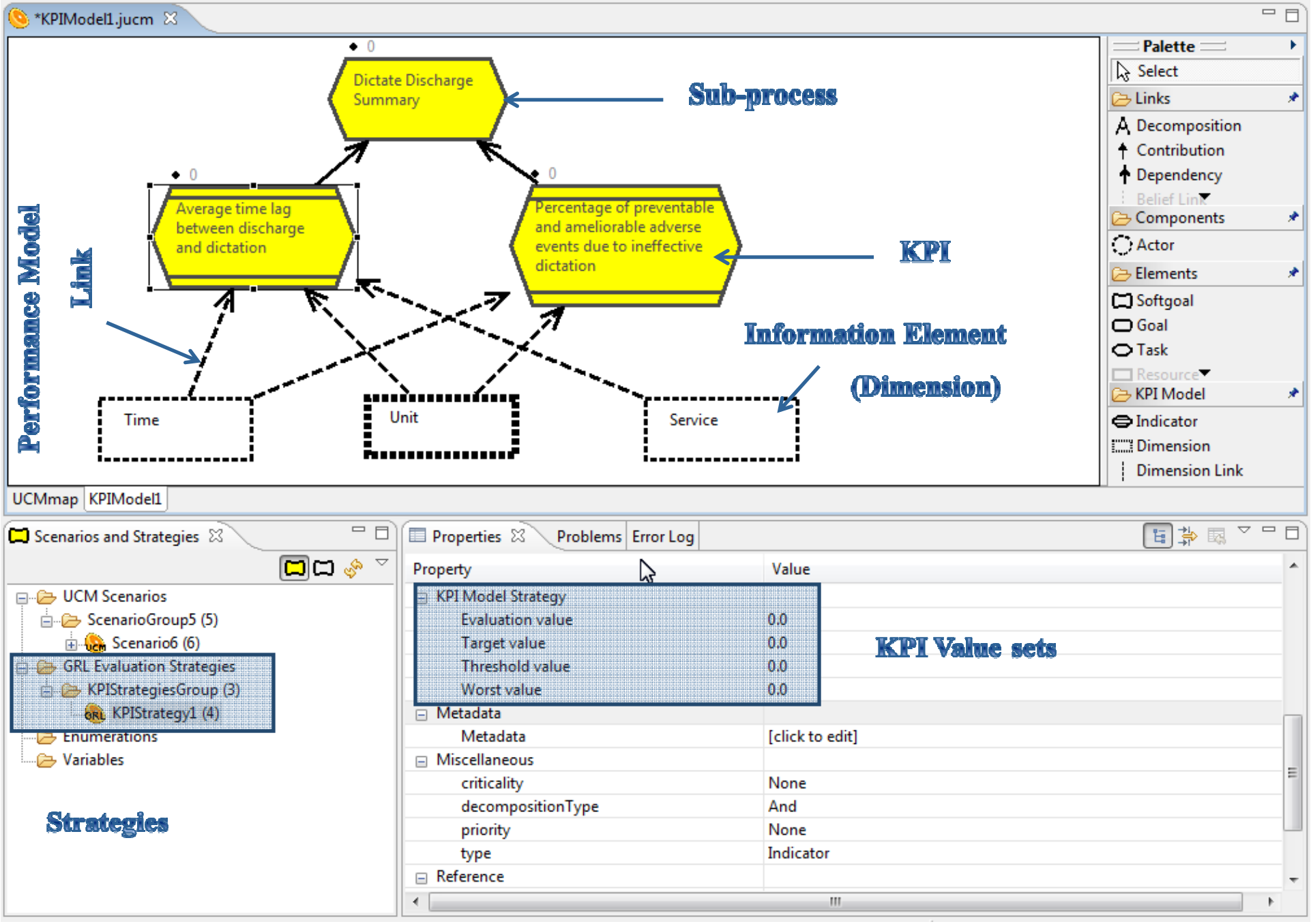


Inclusion of Measures in Goal Models

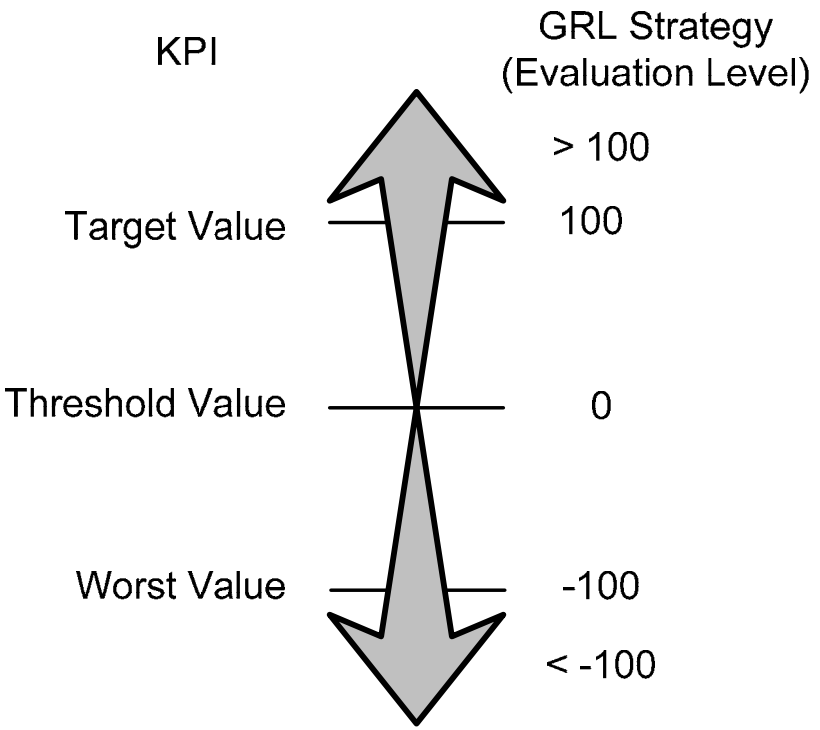
- GRL includes a notion of goal satisfaction, with qualitative and quantitative ([-100..100]) scales.
- However, there is often a need to better relate observations about the real world to the goal model, with domain-specific units such as:
  - Currencies (e.g., revenues in $)
  - Durations (e.g., waiting time in a hospital, in hours)
  - Counts (e.g., number of new students admitted in SEG)
- GRL has non-standard extensions to support this kind of information, and integrate it in the rest of the goal model
  - *Key Performance Indicator* (KPI)
- KPIs help **measure** goals and NFRs with quantifiable metrics
- GRL KPIs can also be fed from external sources of information, hence turning the GRL model into a monitoring engine (e.g., a dashboard).

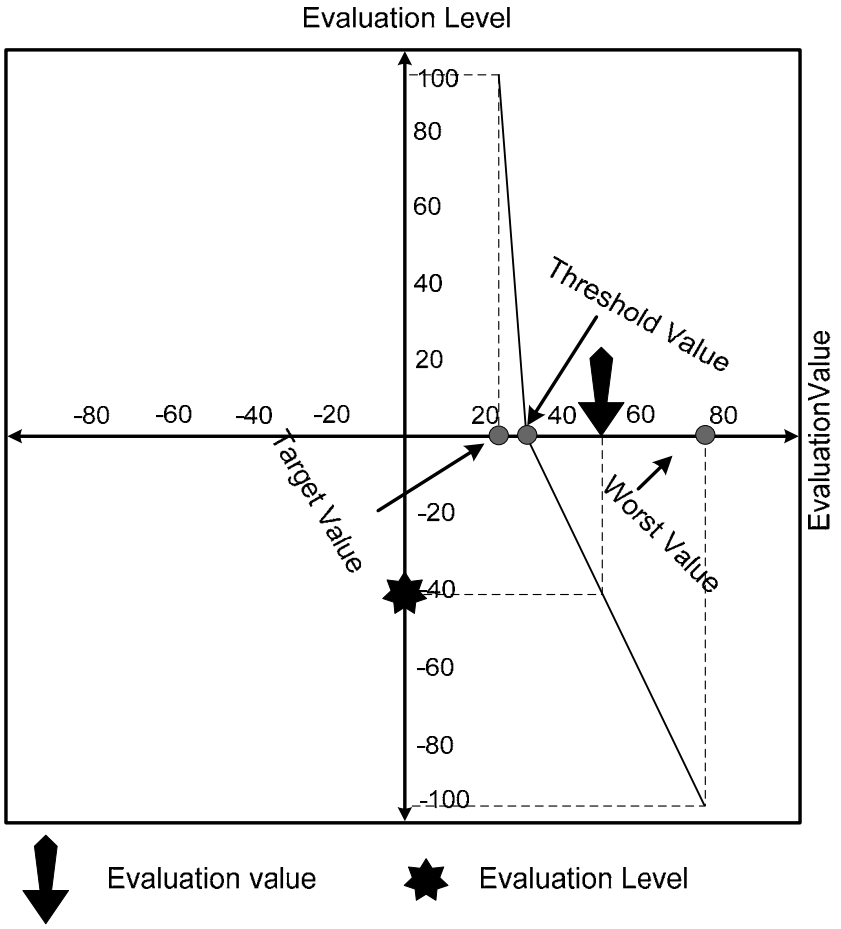
GRL Key performance Indicators

- In GRL, a KPI is defined as an intentional element, but with additional characteristics
  - Attributes (for a given GRL strategy)
    - An **evaluation** value (observed, or simulated in a what-if strategy)
    - A **target** value (the KPI is fully satisfied if the evaluation value reaches it)
    - A **worst-case** value (the KPI is fully denied if the evaluation value reaches it)
    - A **threshold** value (the KPI is neutral if the evaluation value equals it)
    - A **unit** (e.g., $)
  - Associations (for a given GRL model)
    - Can be part of contributions or decompositions
    - New: can be analysed from multiple **dimensions**
      - For example, a *time* dimension might enable the study of a KPI according to a year, a month, a week, a day, or an hour.


*KPIModel1.jucm
Dictate Discharge Summary
Sub-process
Average time lag between discharge and dictation
Percentage of preventable and ameliorable adverse events due to ineffective dictation
KPI
Performance Model
Link
Information Element
(Dimension)
Time
Unit
Service
Palette
Select
Links
Decomposition
Contribution
Dependency
Components
Actor
Elements
Softgoal
Goal
Task
KPI Model
Indicator
Dimension
Dimension Link
UCMmap
KPIModel1
Scenarios and Strategies
UCM Scenarios
ScenarioGroup5 (5)
Scenario6 (6)
GRL Evaluation Strategies
KPIStrategiesGroup (3)
KPIStrategy1 (4)
Enumerations
Variables
Strategies
Properties
Problems
Error Log
Property
Value
KPI Model Strategy
Evaluation value
0.0
Target value
0.0
Threshold value
0.0
Worst value
0.0
KPI Value sets
Metadata
Metadata
[click to edit]
Miscellaneous
criticality
None
decompositionType
And
priority
None
type
Indicator
Reference

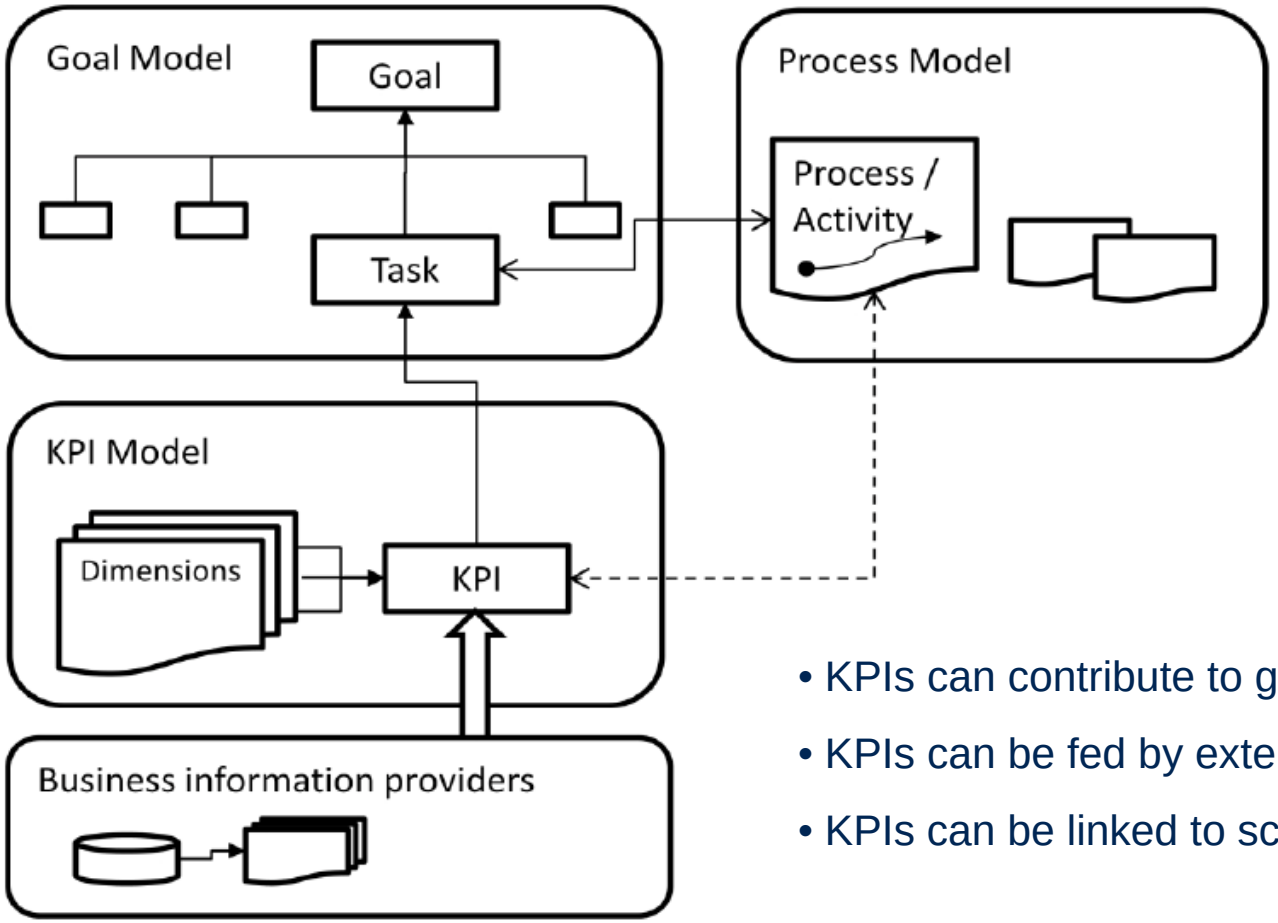
From KPIs to GRL Satisfaction Levels





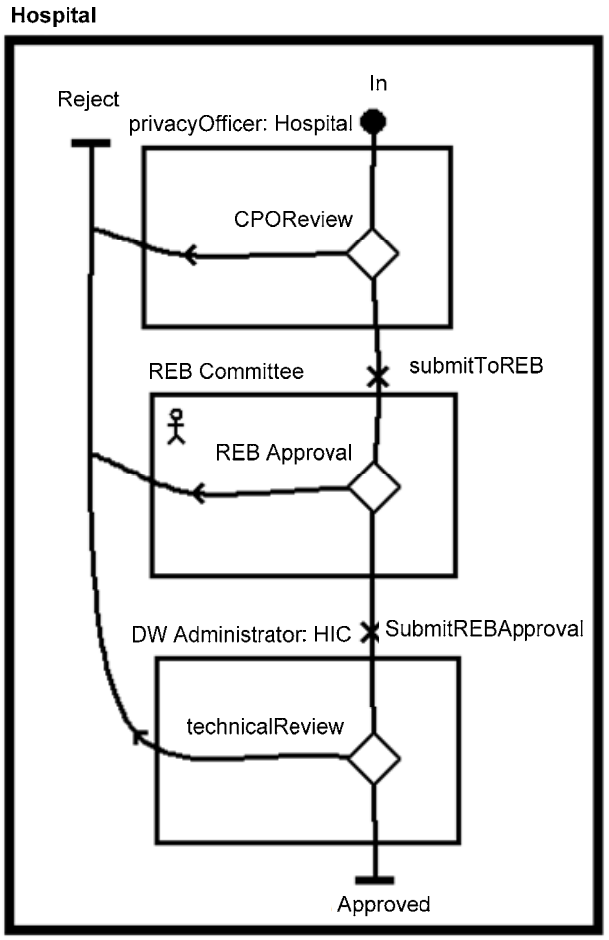
Note: Linear interpolation is currently being used to compute the satisfaction, which is a function of the evaluation, target, threshold, and worst values.

Evaluations Involving KPIs

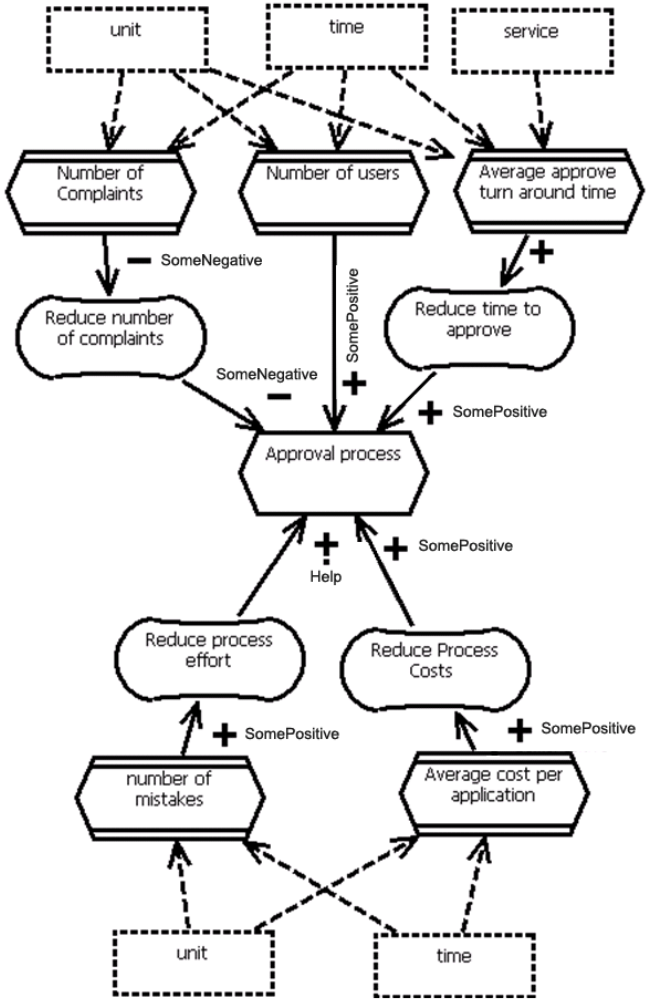


- KPIs can contribute to goals in GRL
- KPIs can be fed by external sources
- KPIs can be linked to scenarios

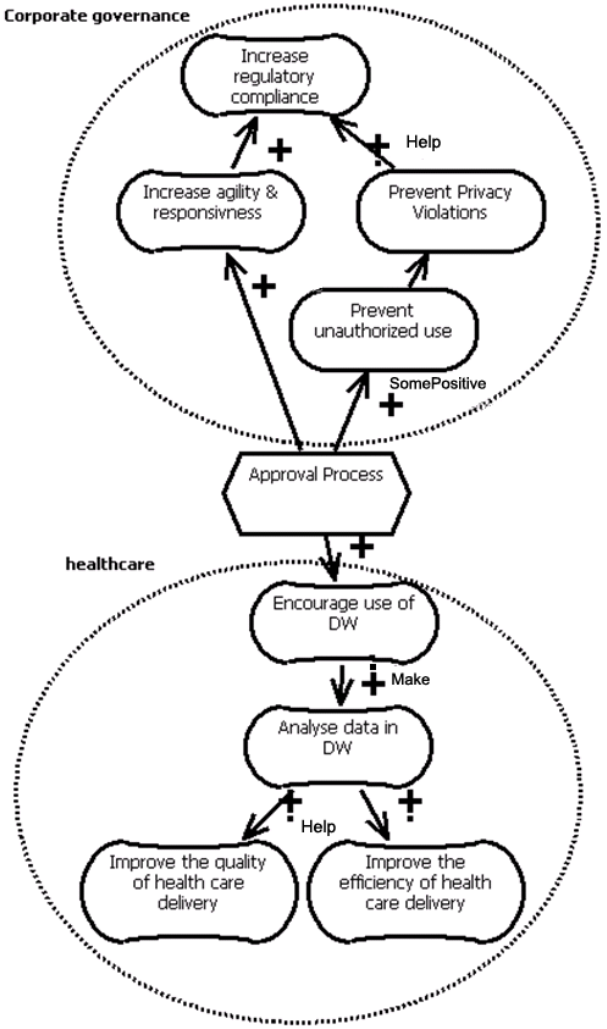
Multiple Views



Process Model

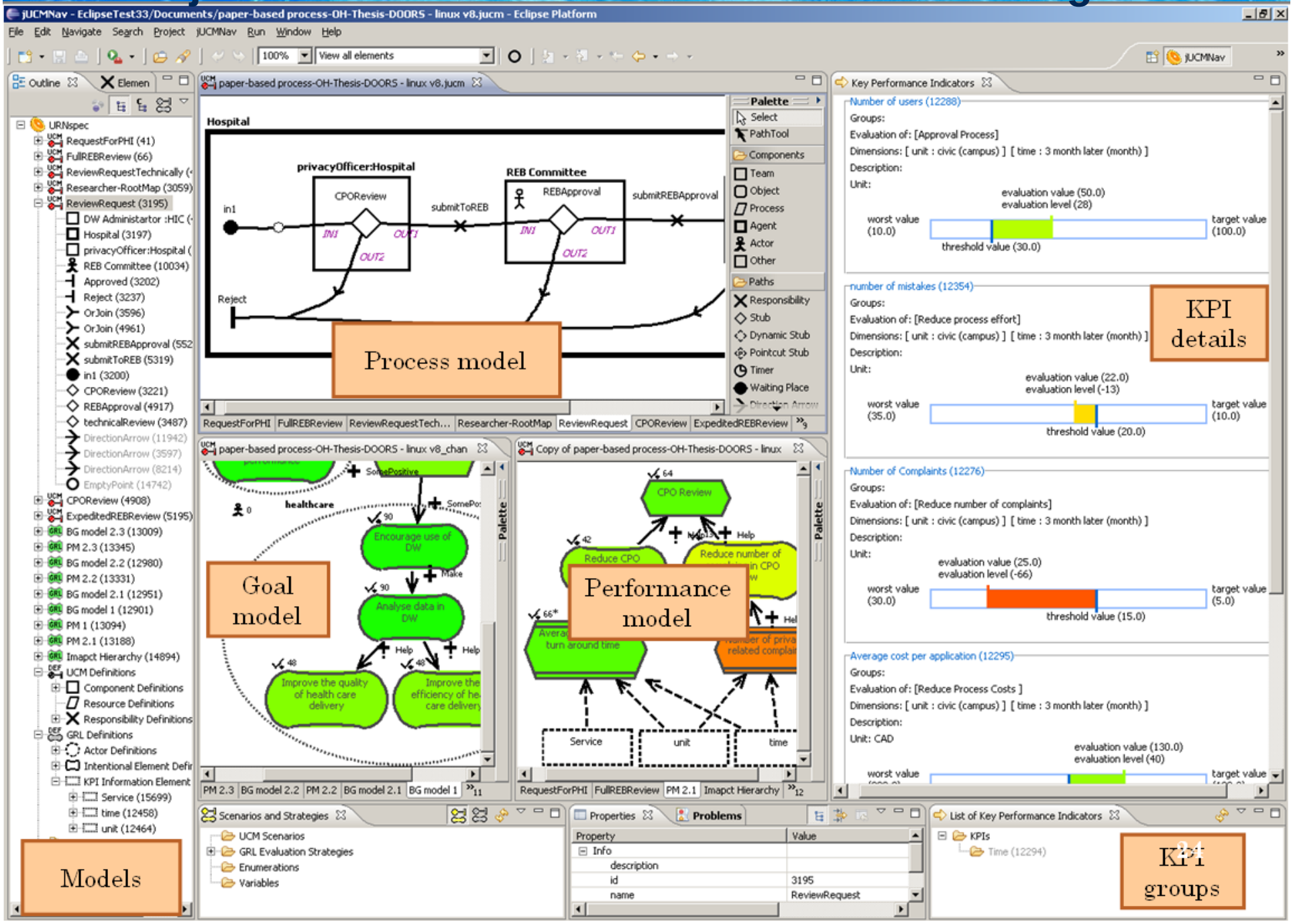


Performance Model



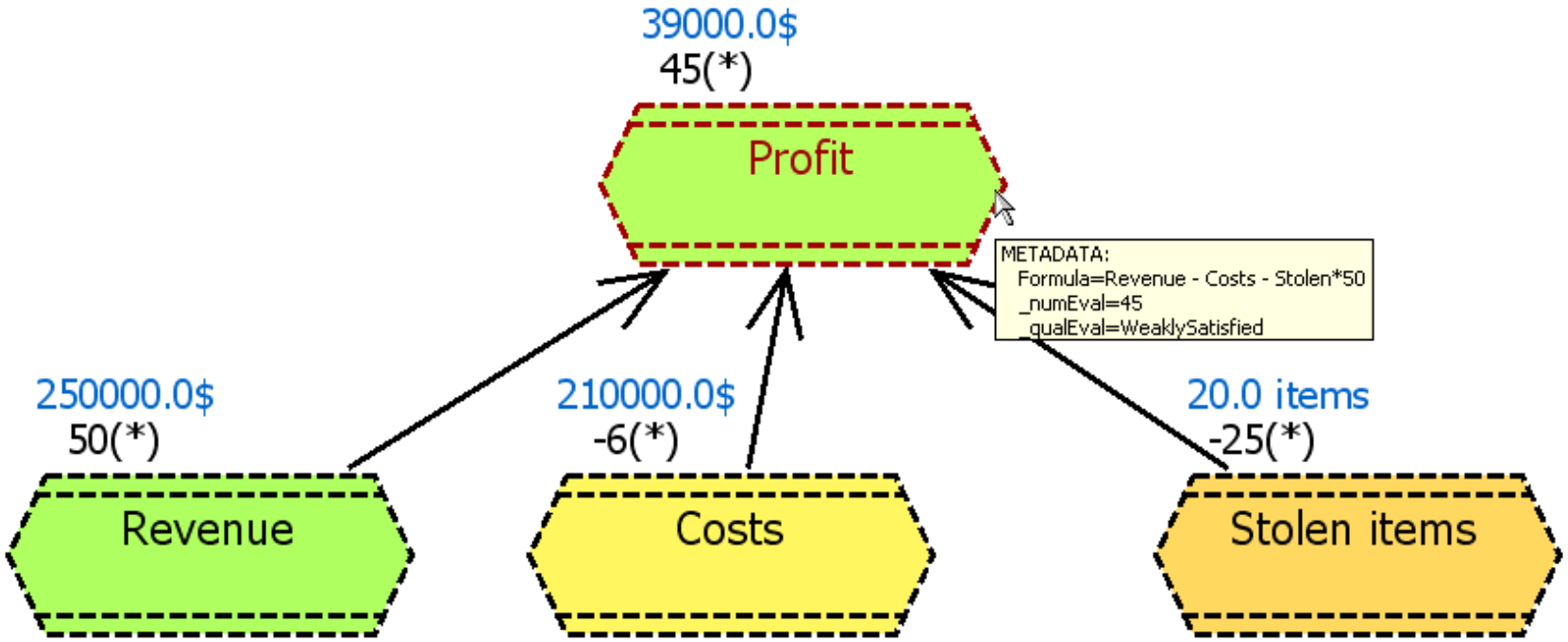
Goal Model

jUCMNav Extensions for KPI and Monitoring



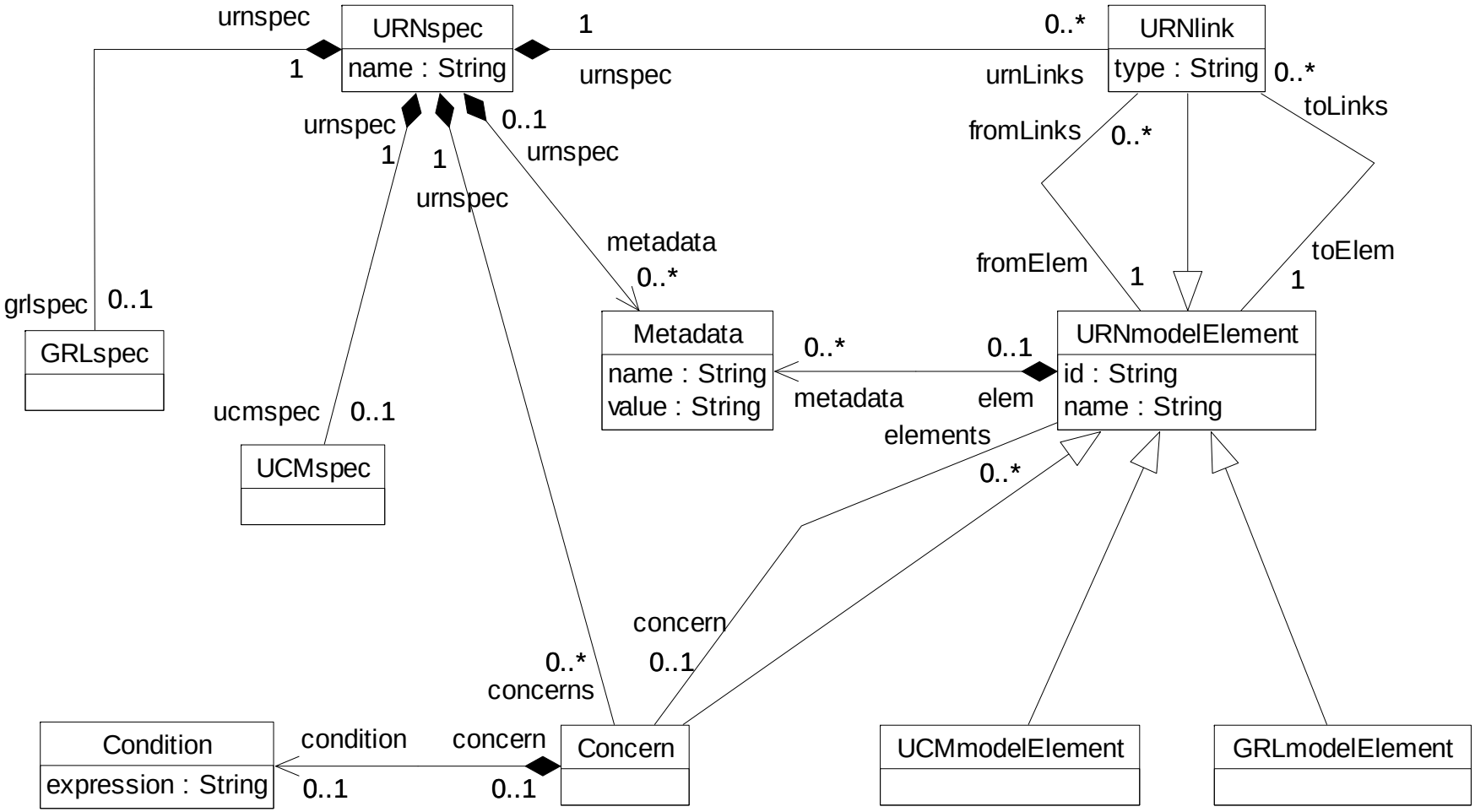
KPI Aggregation

In jUCMNav, KPI values can also be computed (aggregated) from other KPIs, in a way similar to Excel.

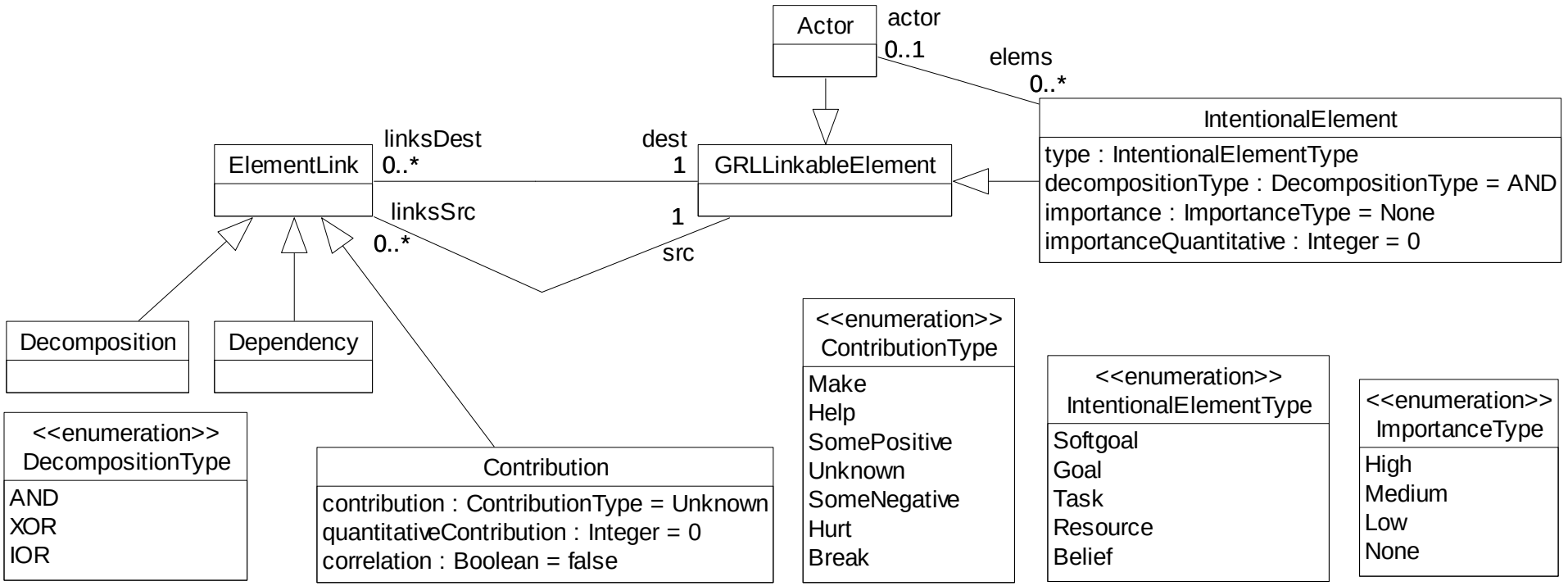


Formula-based GRL evaluation algorithm

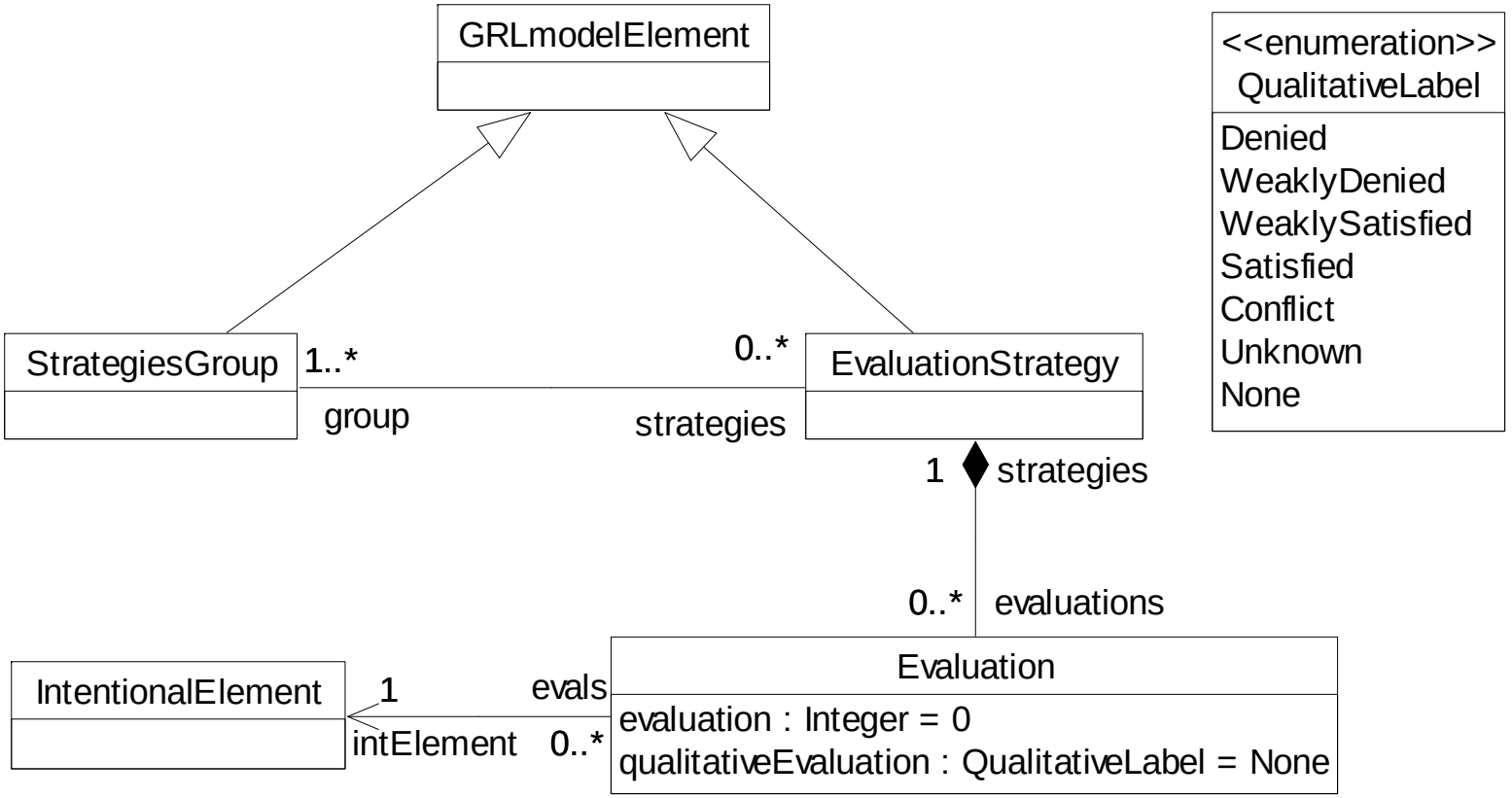
URN Abstract Metamodel



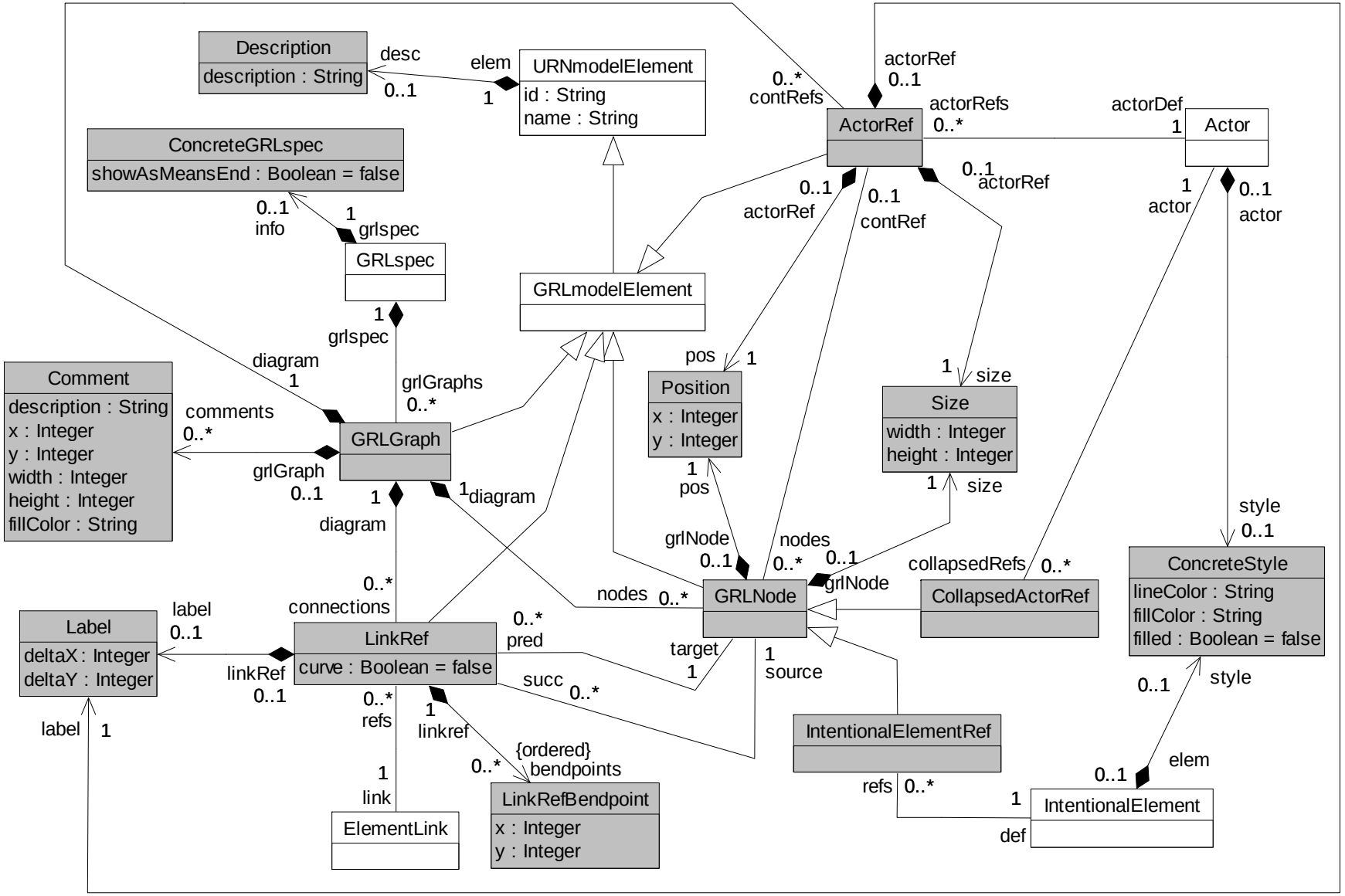
GRL Abstract Metamodel



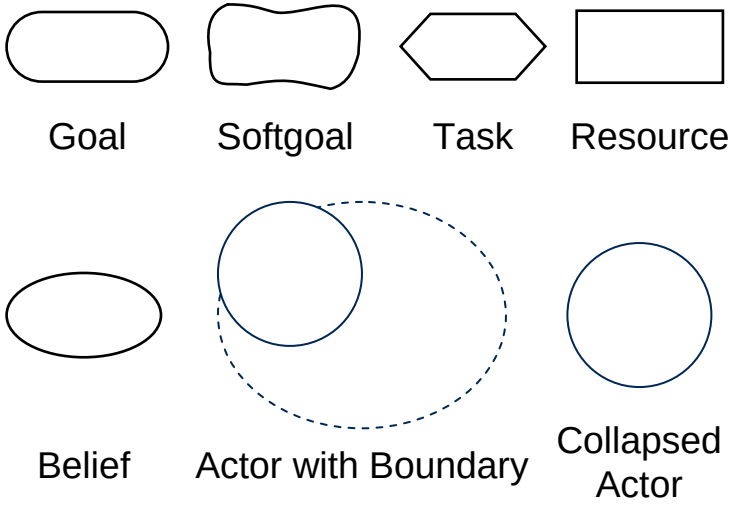
GRL Strategies Abstract Metamodel



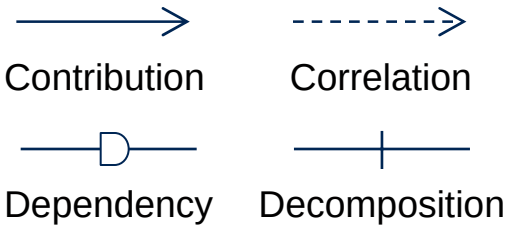
Concrete GRL Metamodel (for Diagrams)



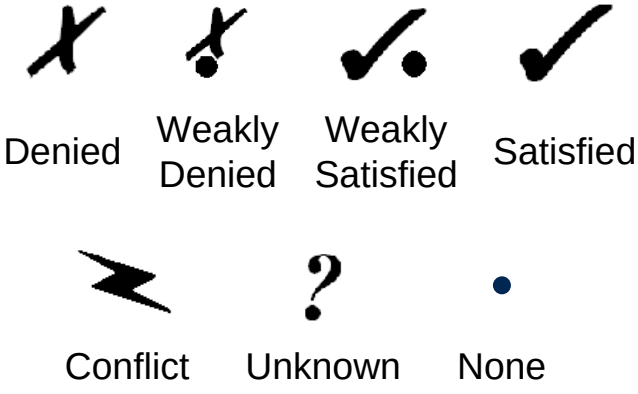
Appendix: GRL Notation



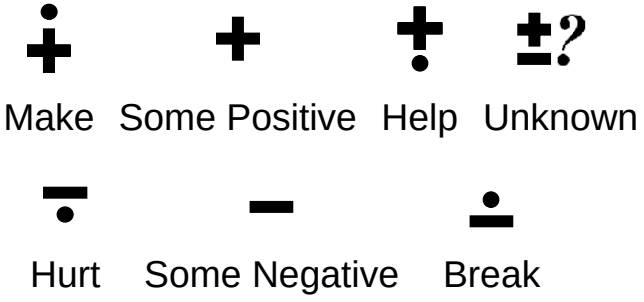
(a) GRL Elements



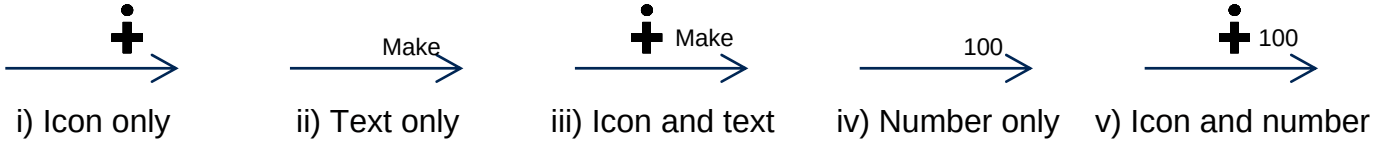
(b) GRL Links



(c) GRL Satisfaction Levels



(d) GRL Contributions Types



(e) Representations of Qualitative and Quantitative Contributions