

CaRE

**A Refinement *Ca*lculus for *Re*quirements *En*gineering
based on Argumentation Theory**

Y. Elrakaiby*, A. Borgida, A. Ferrari, J. Mylopoulos

Lifecycle of Software Projects...!

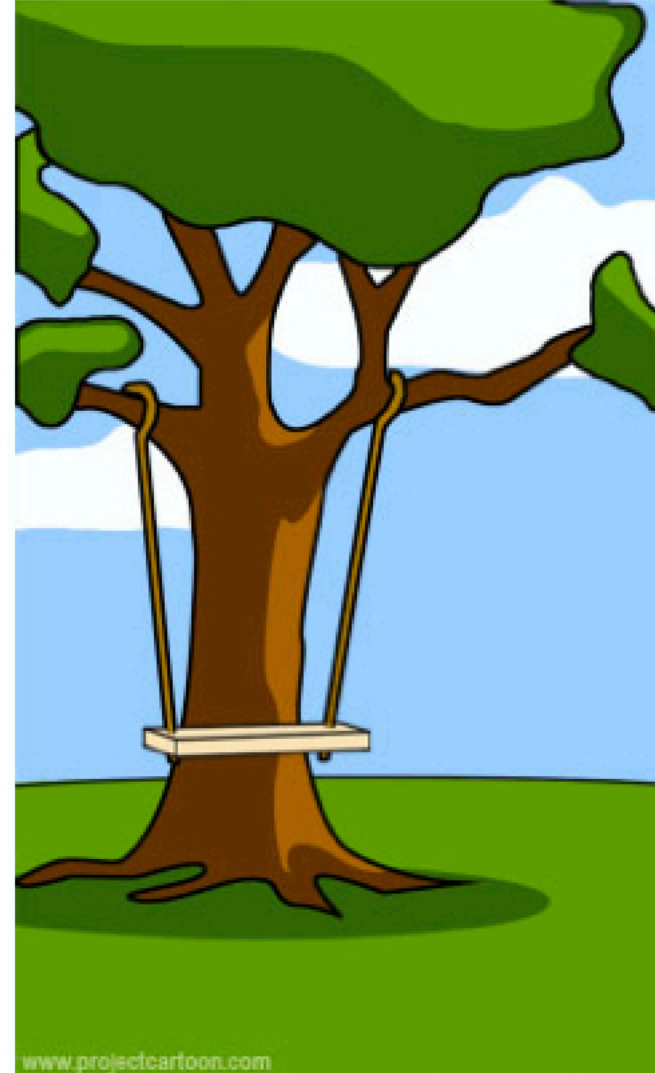
Unfortunately...



What the **customer** needed



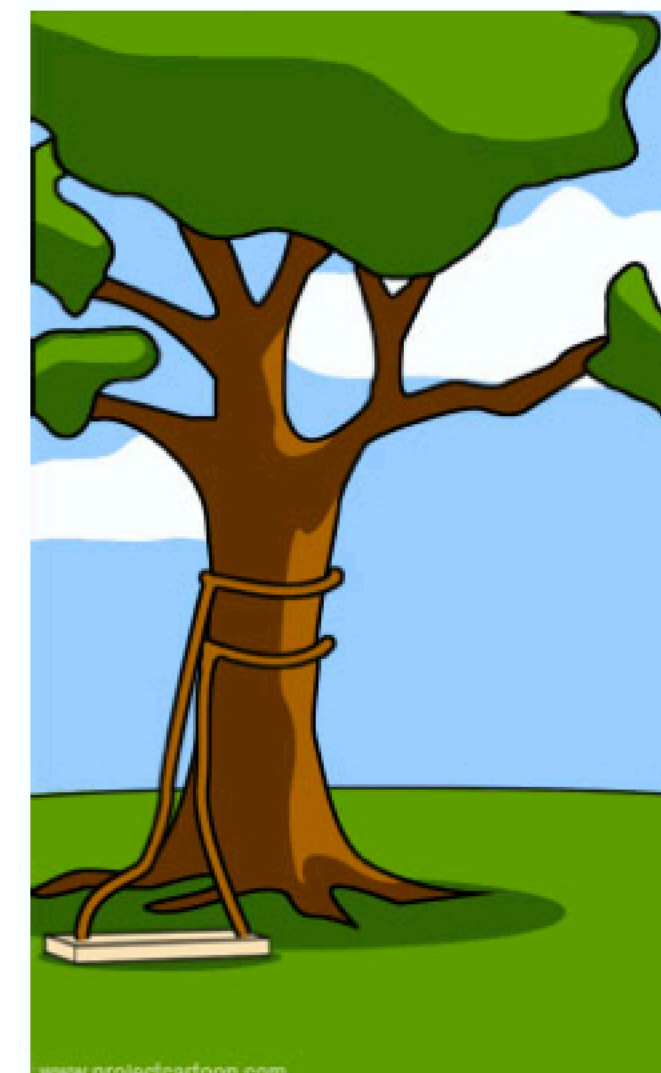
How the **project manager** understood it



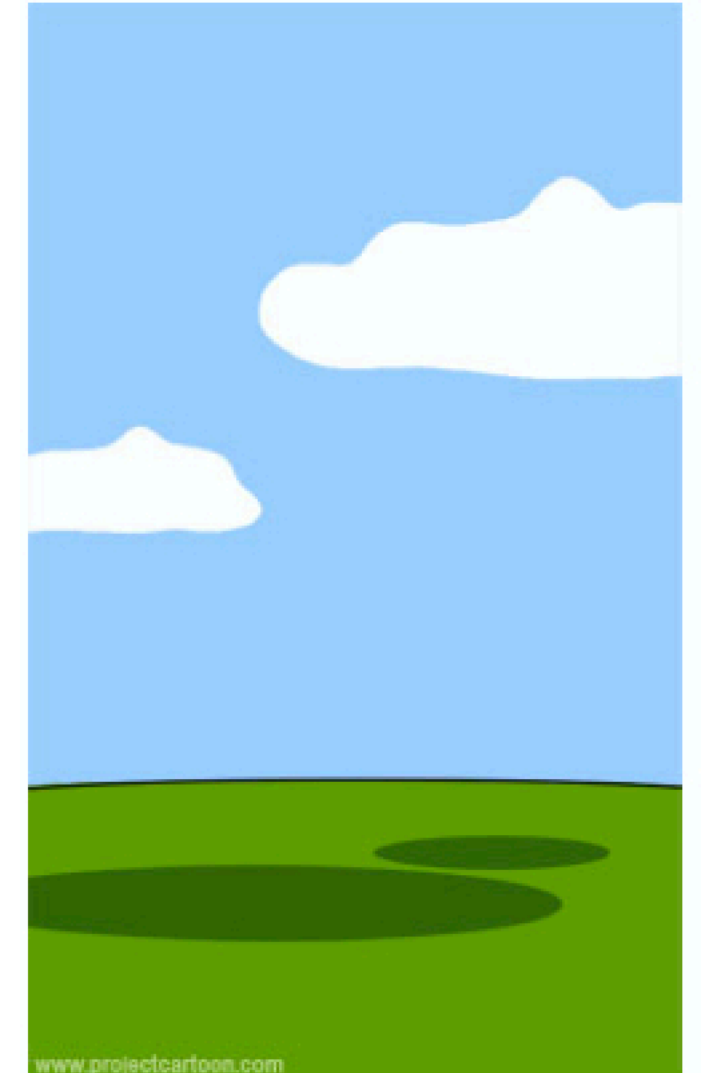
How they explained it



How the **analyst** designs it



How the **developer** developed it



How the project was documented

Solution

Requirements Engineer!

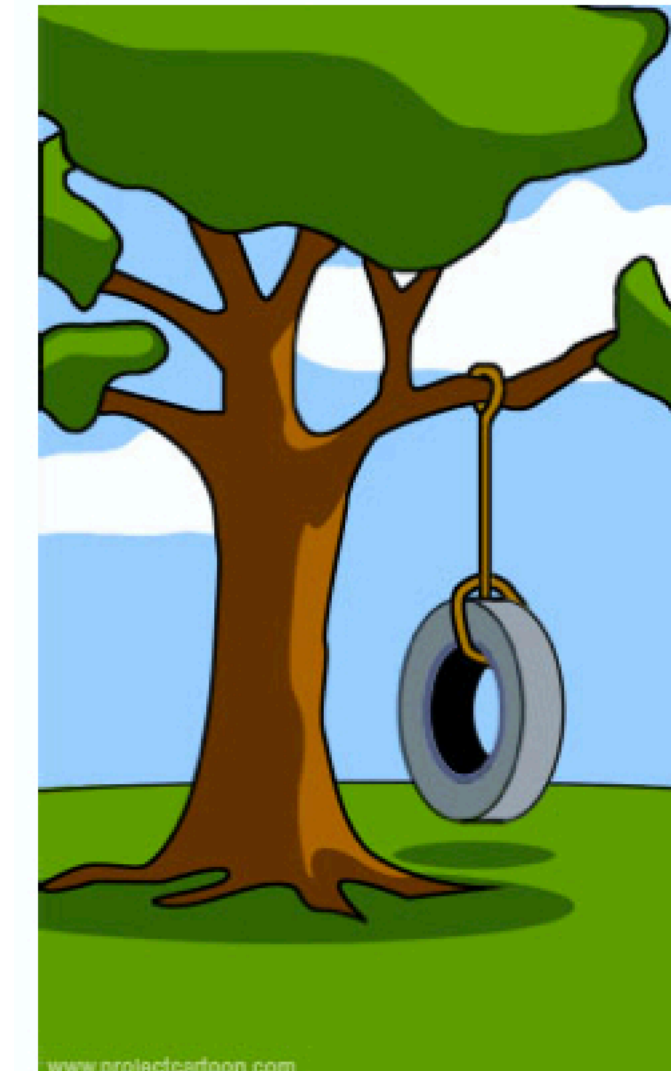


What the customer needed

Requirements Engineer



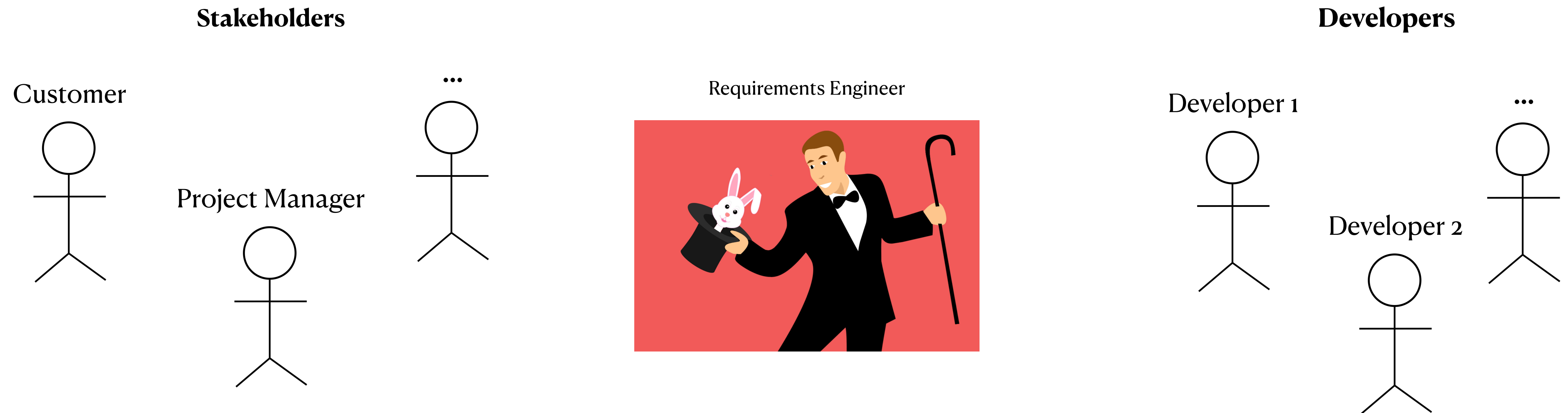
How the developer developed it



How the project was documented

Requirements Engineering

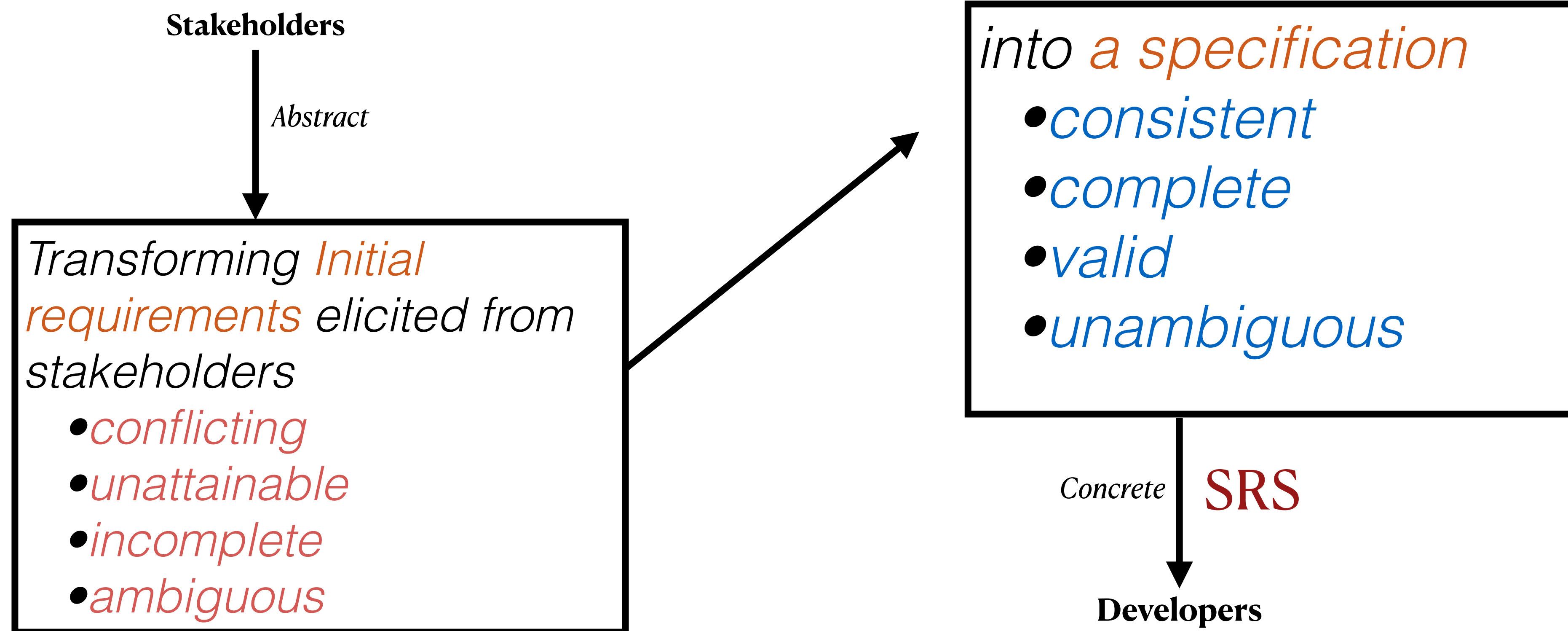
*“Requirements engineering (RE)^[1] is the process of **defining, documenting, and maintaining** requirements^[2].”*



(1) Nuseibeh, B.; Easterbrook, S. (2000). Requirements engineering: a roadmap (PDF). ICSE'00. Proceedings of the conference on the future of Software engineering. pp. 35–46.
(2) Kotonya, Gerald; Sommerville, Ian (September 1998). Requirements Engineering: Processes and Techniques. John Wiley & Sons. ISBN 978-0-471-97208-2.

Our Focus

The *Refinement* Process



“A software requirements specification (SRS) is a document that captures *complete description* about *how the system is expected to perform.*”

Other Challenges

Support of...

- *Negotiation* and *agreement* between stakeholders
- *Documentation*
 - Keep track of rationale of design choices, defects identified in requirements and refinements proposed to address identified defects
- *Change Management*

Requirements Engineering in the Literature

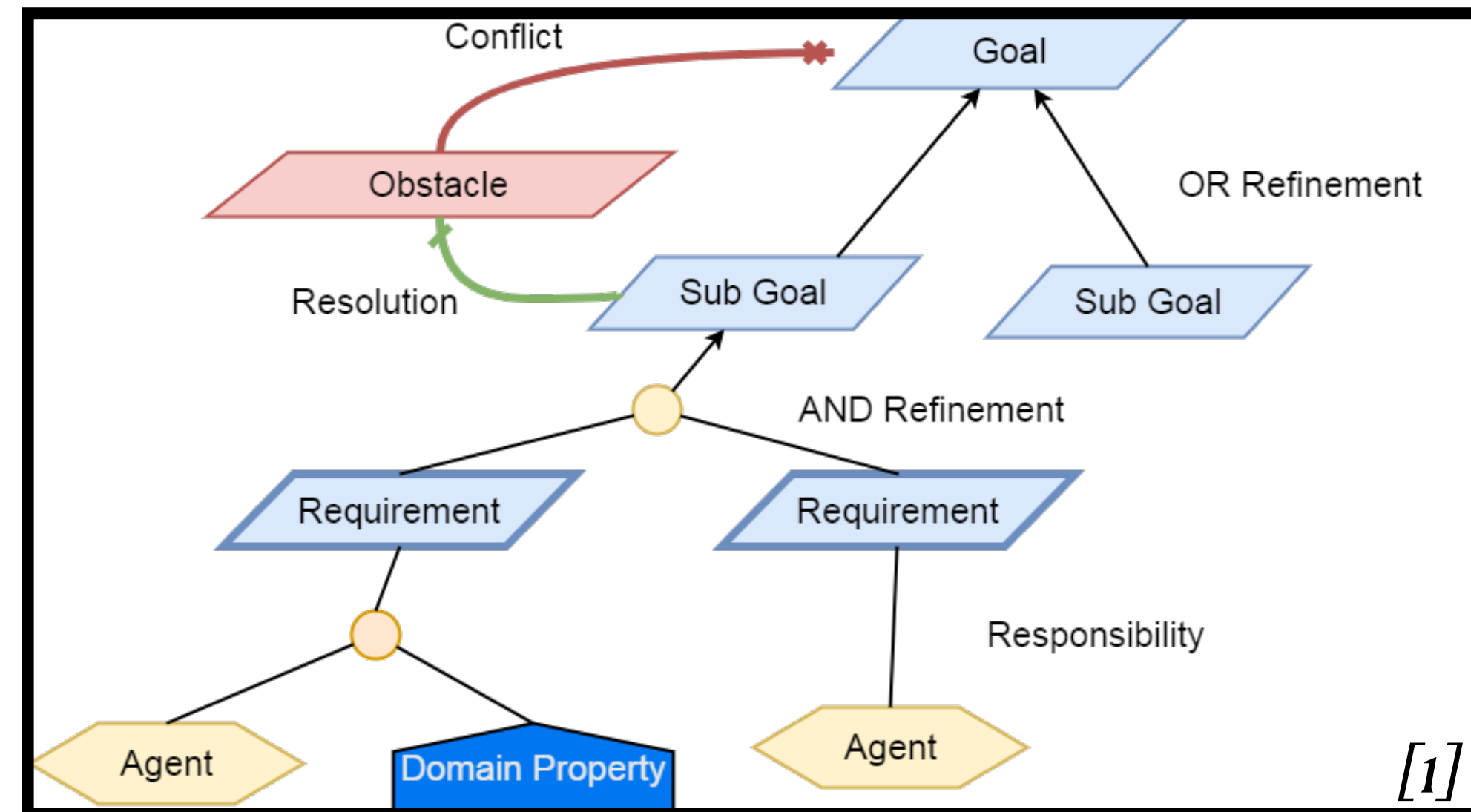
Requirements Specification Methodologies

- Generally revolves around refinement, which have taken many forms:
 - activity decomposition (Ross),
 - abductive inference (Jackson),
 - *goal refinement* (vanLamsweerde),
 - social delegation (Yu).
- Requirements Specification Languages
 - Out of scope

Goal Decomposition in GORE

Example: KAOS Goal Decomposition (Simplified)

- *And/Or* Goal Decomposition
- Leaves are
 - *Tasks* (implementation activities)
 - *Domain Assumptions/Properties*
- Formal Semantics
 - *First-order Temporal Logic*



(1) https://opnfv-availability.readthedocs.io/en/latest/development/overview/HA_Analysis-Gambia.html

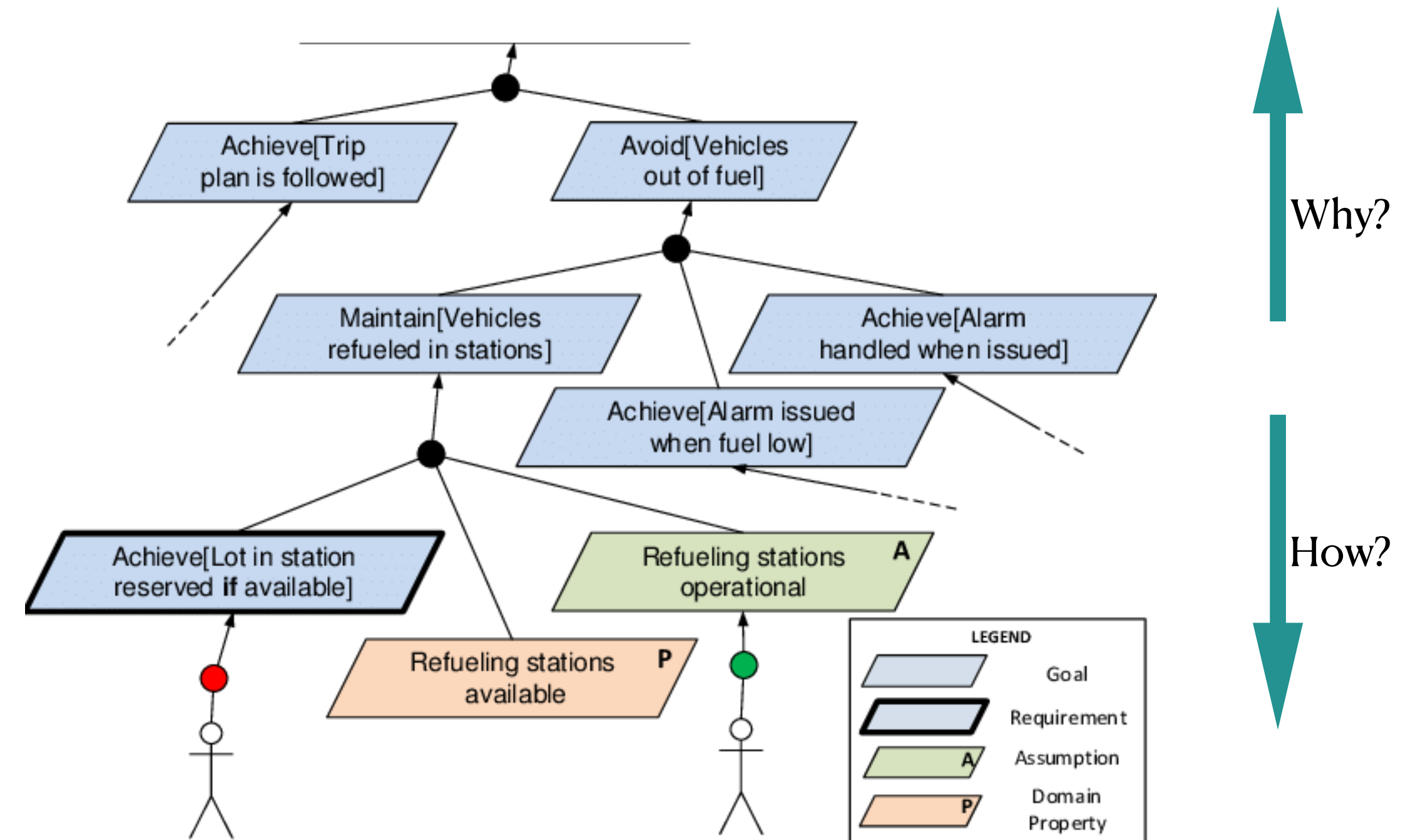
(2) <https://www.cse.msu.edu/~cse870/Materials/GoalModeling/KaosTutorial-2007.pdf>

(3) <http://www.objectiver.com/fileadmin/download/documents/generalleaflet.pdf>

Goal Decomposition in GORE

Example: KAOS Goal Decomposition (Simplified)

- *And/Or* Goal Decomposition
- Leaves are
 - *Tasks* (implementation activities)
 - *Domain Assumptions/Properties*
- Formal Semantics
 - *First-order Temporal Logic*



- (1) <https://www.cse.msu.edu/~cse870/Materials/GoalModeling/KaosTutorial-2007.pdf>.
- (2) <http://www.objectiver.com/fileadmin/download/documents/generalleaflet.pdf>.

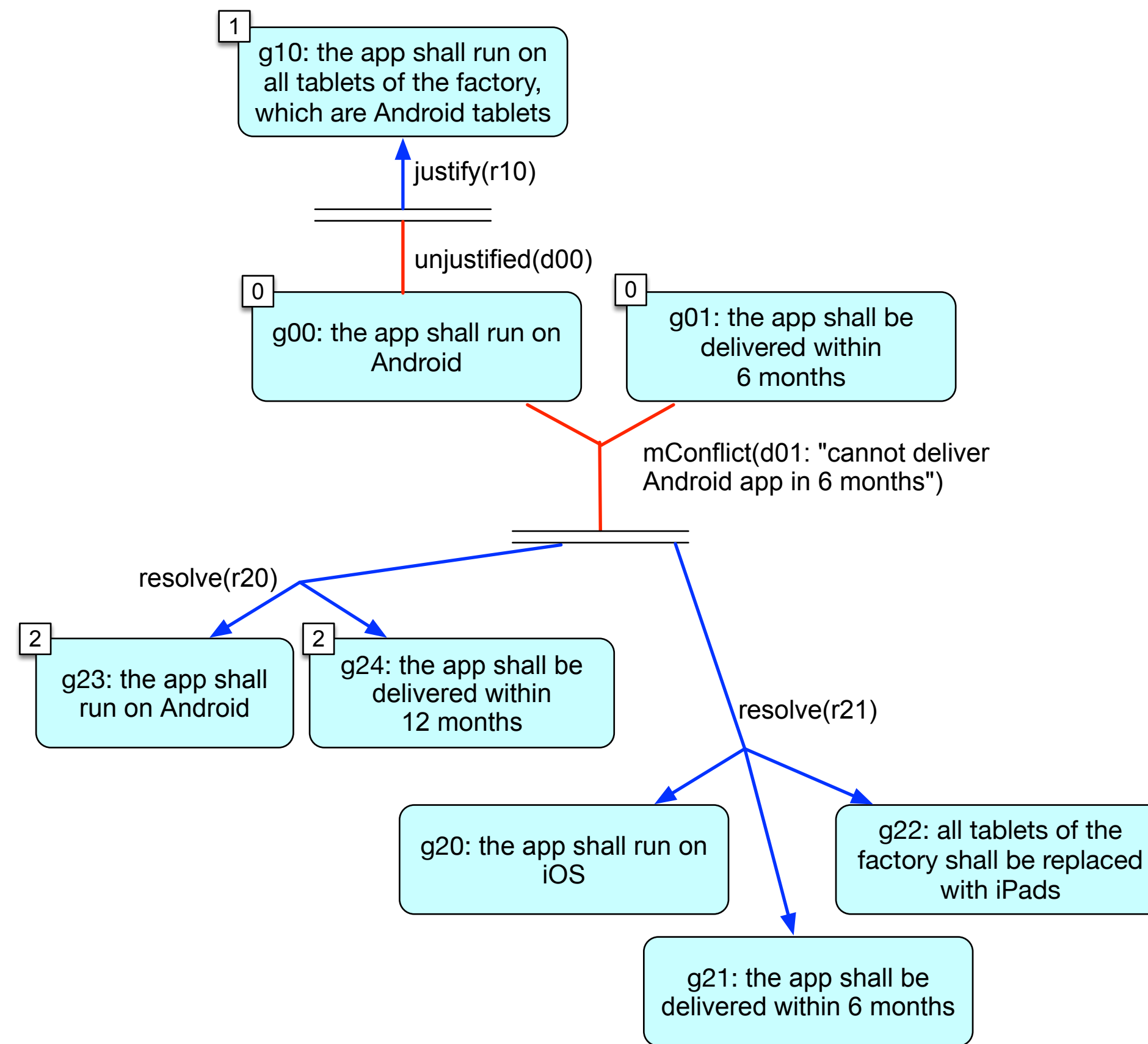
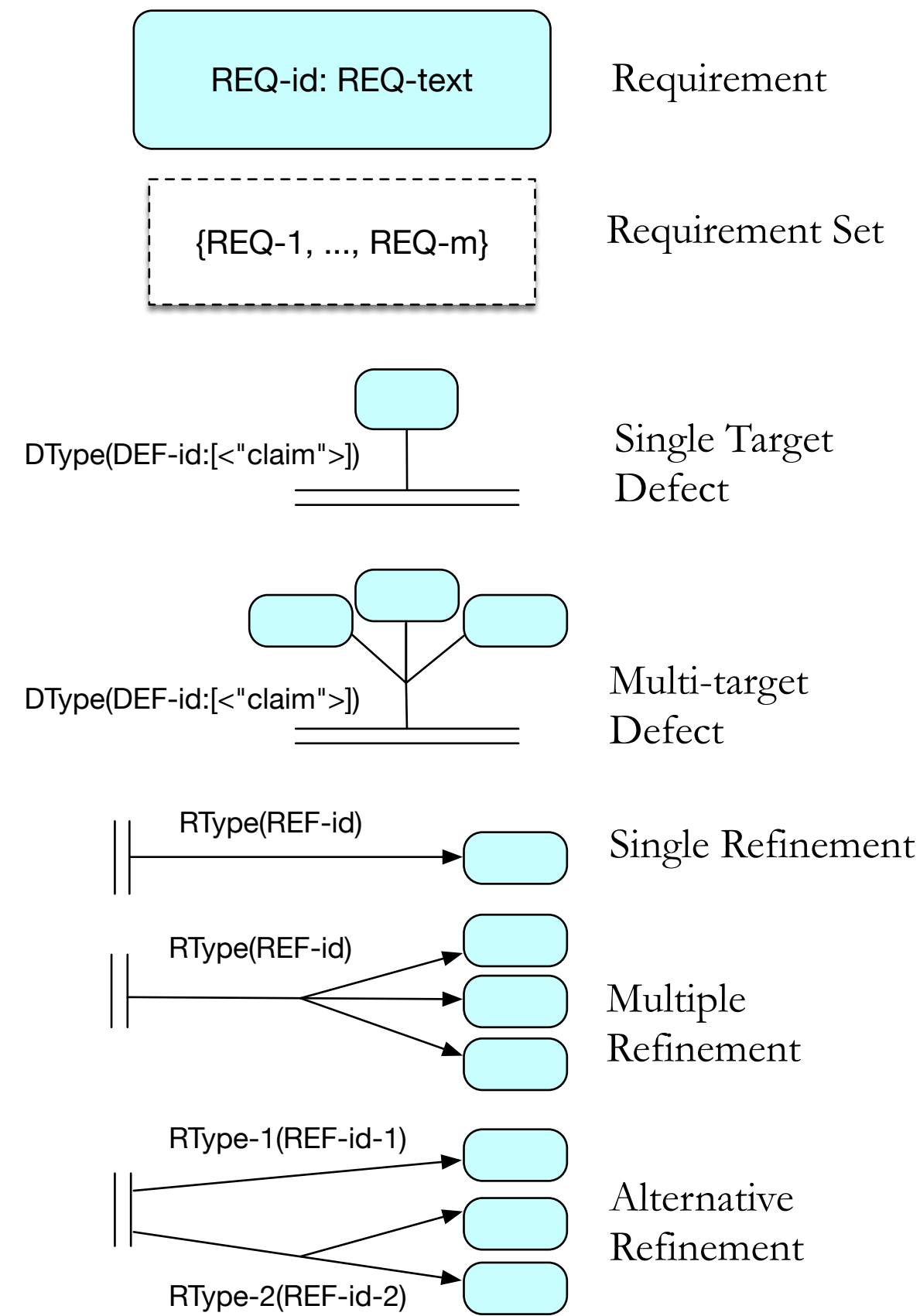
CaRE

A Calculus for Requirements Engineering

- *Defects* are first-class entities
 - Keep track of identified defects, hence document the rational behind refinements and evolution of the refinement process
- *Refinements* to address each defect type
- Formalised using *Argumentation Theory*
 - Support convergence and agreement between stakeholders
 - Reasoning support
- *Tool* support

CaRE

The Graphical Notation

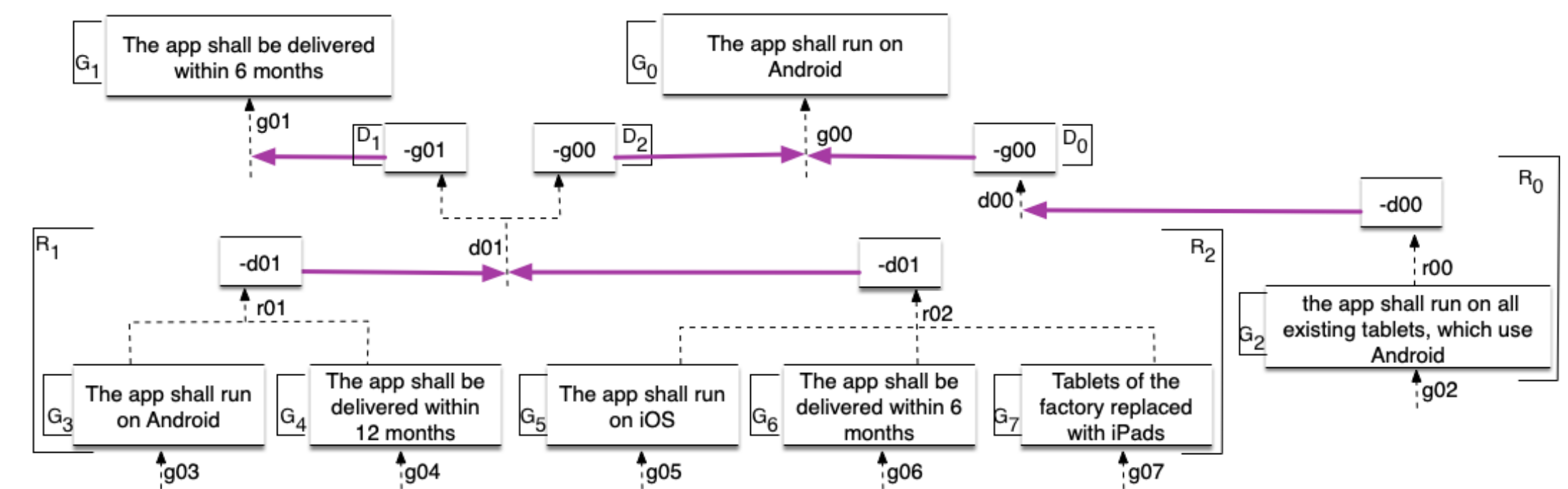


CaRE

The Semantics

- *Formal semantics* in terms of ASPIC⁺ Argumentation Theory
 - Reasoning about arguments and conflicts between them
 - Computation of extensions

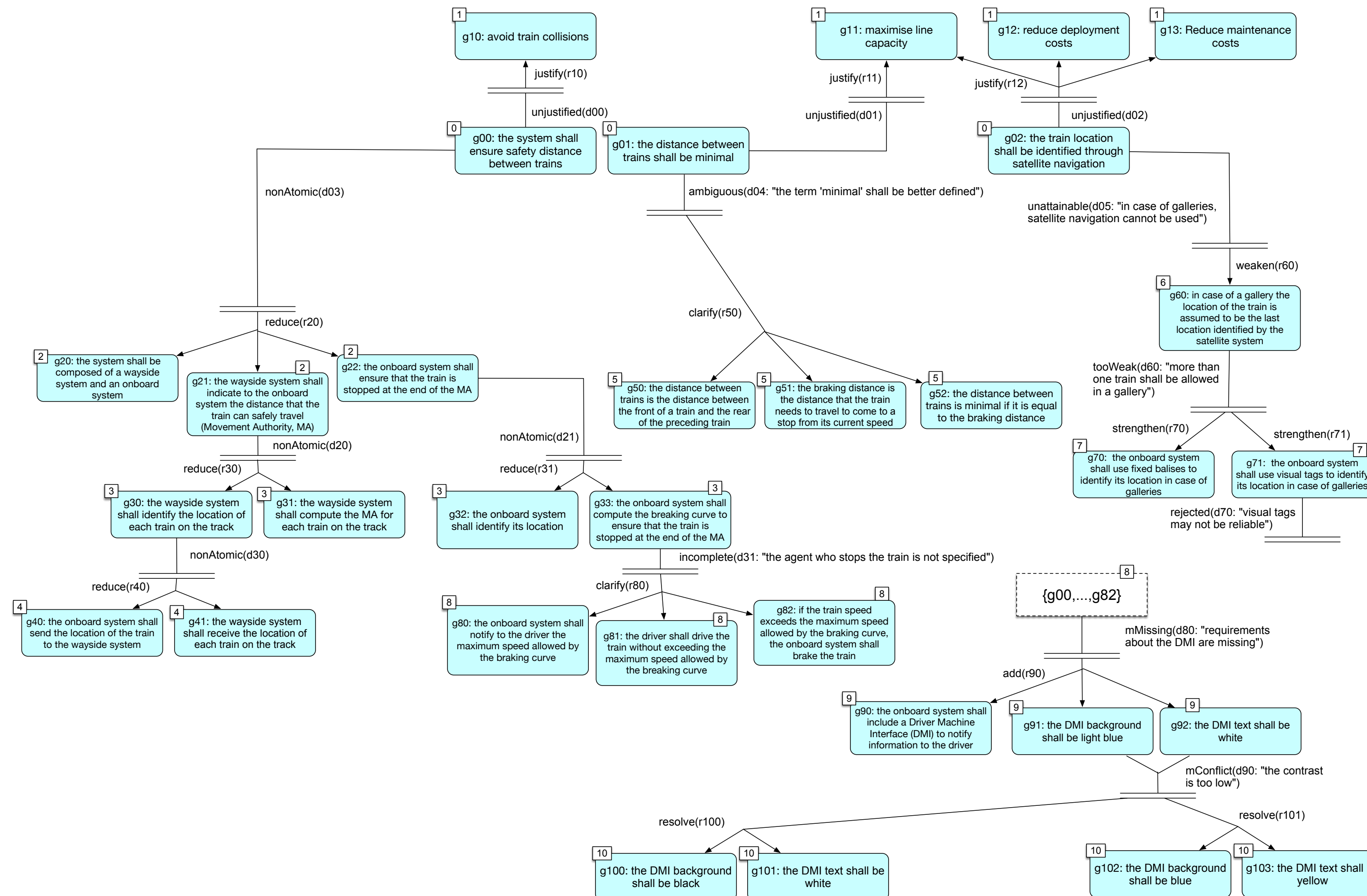
- A set of *arguments*
- *Conflict* relations



- Sets of *arguments* which may be *collectively accepted*

CaRE

Tool Support: Example



CaRE

Tool Support

- Implemented in Java
- Runs through the command line using a textual description of refinement graphs
- Reasoning tasks
 - Determine *acceptability* of the initial requirements and refinements
 - Compute *minimal specifications*
 - *Minimal set of leaf requirements that make the initial requirements acceptable*

```
INFO: File ./RefGraphs/exampleApplicationScenario.pl read
INFO: Total Number of Requirements is 32
INFO: Initial Requirements (number of requirements is 3)
g00:'the system shall ensure safety distance between trains'
g01:'the distance between trains shall be minimal'
g02:'the train location shall be determined through satellite navigation'
INFO: Leaf (Specification) Requirements (number of requirements is 17)
g100:'the DMI background shall be black'
g101:'the DMI text shall be while'
g102:'the DMI background shall be blue'
g103:'the DMI text shall be yellow'
g20:'the system shall be composed of a wayside system and an onboard system'
g31:'the wayside system shall compute the MA for each train on the track'
g32:'the onboard system shall identify its location'
g40:'the onboard system shall send the location of the train to the wayside system'
g41:'the wayside system shall receive the location of each train on the track'
g50:'the distance between trains is the distance between the front of a train and the rear of the preceding train'
g51:'the bracing distance is the distance that the train needs to travel to come to a full stop from its current speed'
g52:'the distance between trains is minimal if it is equal to the braking distance'
g70:'the onboard system shall use fixed ballises to identify its location in case of galleries'
g80:'the onboard system shall notify the driver of the maximum speed allowed by the braking curve'
g81:'the driver shall drive the train without exceeding the maximum speed allowed by the braking curve'
g82:'if the train speed exceeds the maximum speed allowed by the braking curve the onboard system shall brake the train'
g90:'the onboard system shall include a Driver Machine Interface DMI to notify information to the driver'

Printing Minimal Specs
Minimal Specification 1 (number of requirements is 15)
g102:'the DMI background shall be blue'
g103:'the DMI text shall be yellow'
g20:'the system shall be composed of a wayside system and an onboard system'
g31:'the wayside system shall compute the MA for each train on the track'
g32:'the onboard system shall identify its location'
g40:'the onboard system shall send the location of the train to the wayside system'
g41:'the wayside system shall receive the location of each train on the track'
g50:'the distance between trains is the distance between the front of a train and the rear of the preceding train'
g51:'the bracing distance is the distance that the train needs to travel to come to a full stop from its current speed'
g52:'the distance between trains is minimal if it is equal to the braking distance'
g70:'the onboard system shall use fixed ballises to identify its location in case of galleries'
g80:'the onboard system shall notify the driver of the maximum speed allowed by the braking curve'
g81:'the driver shall drive the train without exceeding the maximum speed allowed by the braking curve'
g82:'if the train speed exceeds the maximum speed allowed by the braking curve the onboard system shall brake the train'
g90:'the onboard system shall include a Driver Machine Interface DMI to notify information to the driver'

Minimal Specification 2 (number of requirements is 15)
g100:'the DMI background shall be black'
g101:'the DMI text shall be while'
g20:'the system shall be composed of a wayside system and an onboard system'
g31:'the wayside system shall compute the MA for each train on the track'
g32:'the onboard system shall identify its location'
g40:'the onboard system shall send the location of the train to the wayside system'
g41:'the wayside system shall receive the location of each train on the track'
g50:'the distance between trains is the distance between the front of a train and the rear of the preceding train'
g51:'the bracing distance is the distance that the train needs to travel to come to a full stop from its current speed'
g52:'the distance between trains is minimal if it is equal to the braking distance'
g70:'the onboard system shall use fixed ballises to identify its location in case of galleries'
g80:'the onboard system shall notify the driver of the maximum speed allowed by the braking curve'
g81:'the driver shall drive the train without exceeding the maximum speed allowed by the braking curve'
g82:'if the train speed exceeds the maximum speed allowed by the braking curve the onboard system shall brake the train'
g90:'the onboard system shall include a Driver Machine Interface DMI to notify information to the driver'

Computation of minimal specifications time = 362ms
Printing of minimal specifications time = 1ms
Parsing refinement graph time = 19ms
```

Conclusion

Final Thoughts...

	GORE	CaRE
Documentation	Mainly AND/OR decomposition (Nonatomic defect)	More <i>defects and refinements</i>
Change Management	Complex	Simple (monotonic)
Formal Semantics	Yes	Yes
Reasoning support	Yes	Yes
Tool Support	UI	Basic
Support of finding agreement between Stakeholders	No	Yes

Future Work

Perspectives

- Structured *requirement specification language*
- Formalisation of *valid refinements*
- *Relationships* between requirements
 - Equivalence
 - Implication
 - Incompatibility
- *UI* for the tool