

Goal-Oriented Requirements Engineering and Quality Requirements

Nicolas Gordillo & Jarosław Kuśnierz

Goal-Directed Requirements Acquisition

- Goals lead to requirements
- Goals justify and explain requirements to stakeholders
- Goals assign responsibilities to agents
- Goals provide basic information for detecting and resolving conflicts
- Most of these can be achieved by using KAOS.

KAOS(Knowledge Acquisition in autOmated Specification)

By using the model we get Goals, Agents, Concerned Objects, Actions, Constraints and Responsibilities.

KAOS uses modelling patterns to deal with goals, states and their requirements.

Process?

Process

- *Identify Goals, and their Concerned Objects.*
- *Identify potential Agents and their Capabilities.*
- *Operationalize Goals into Constraints.*
- *Refine Objects and Actions.*
- *Identify alternative Responsibilities.*
- *Assign Actions to responsible Agents.*

Iterate

What does KAOS solve?

In requirements engineering, we need to justify the goals by explaining WHY is it needed.

The collection of subgoals describe HOW it can be reached.

Provides with a top-down, bottom-up and non-directional approach to deal with goals dependencies.

“ a complete KAOS model leaves no space for wishful thinking (a goal not refined), requirements with no responsible, unjustified operations, and operations for which we ignore who will execute what and when. ”

KAOS: Automated Train Control System Example

[A. van Lamsweerde, "Requirements engineering in the year 00: a research perspective", *Proc., 22nd ICSE'00*, pp5-19. IEEE Computer Society Press]

Example



Goal Maintain[TrackSegmentSpeedLimit]

InformalDef A train should stay below the maximum speed the track segment can handle.

FormalDef $\forall tr: \text{Train}, s: \text{TrackSegment} :$
 $\text{On}(tr, s) \Rightarrow tr.\text{Speed} \leq s.\text{SpeedLimit}$

Goal Maintain[WCS-DistBetweenTrains]

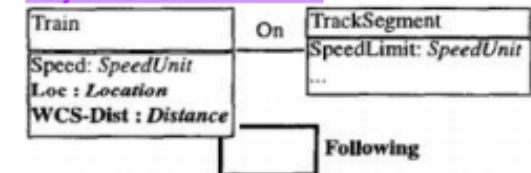
InformalDef A train should never get so close to a train in front so that if the train in front stops suddenly (e.g., derailment) the next train would hit it.

FormalDef $\forall tr1, tr2: \text{Train} :$
 $\text{Following}(tr1, tr2) \Rightarrow tr1.\text{Loc} - tr2.\text{Loc} > tr1.\text{WCS-Dist}$

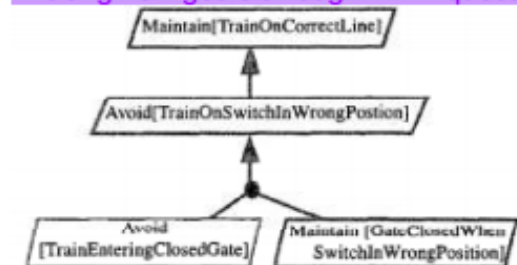
Lawrence Chung

worst case stopping

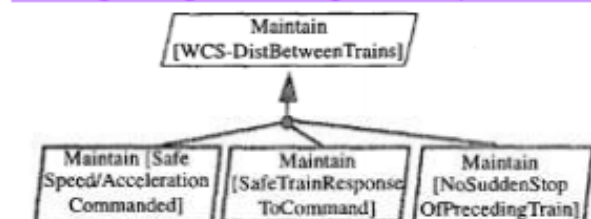
Object model



Eliciting new goals through WHY questions

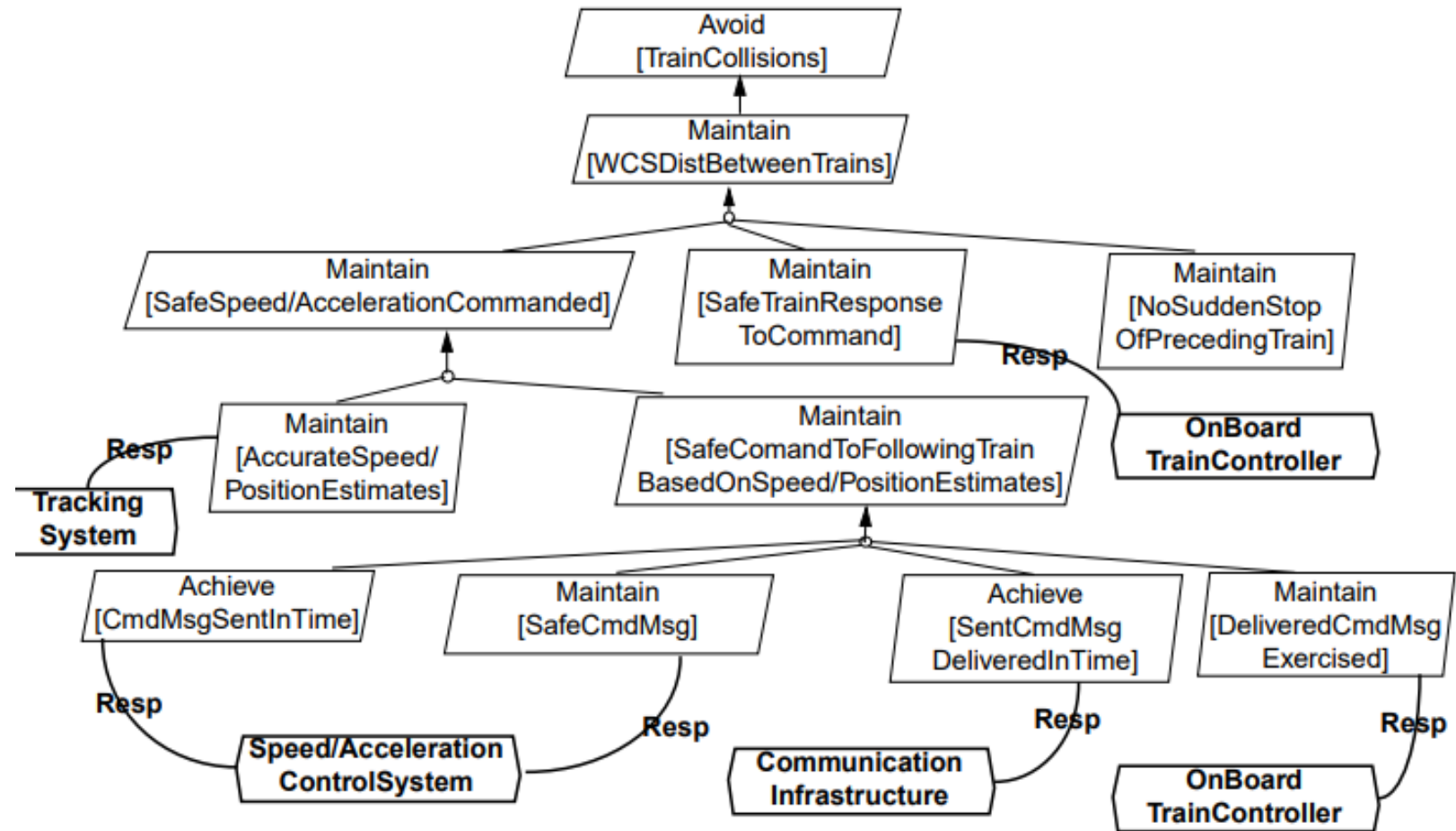


Eliciting new goals through HOW questions



Agents

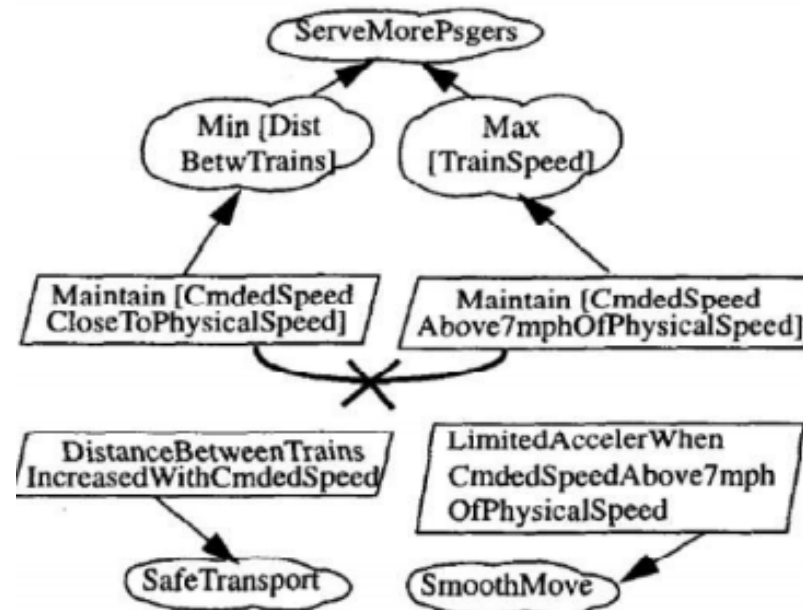
Identifying Potential Responsibility Assignments



KAOS: Automated Train Control System Example

[A. van Lamsweerde, "Requirements engineering in the year 00: a research perspective", Proc., 22nd ICSE'00, pp5-19. IEEE Computer Society Press]

Conflicts



Conflicts

Conflicting Goals:

Goal Maintain [CmdedSpeedCloseToPhysicalSpeed]

FormalDef $\forall tr: Train$
 $tr.Acc_{CM} \geq 0$
 $\Rightarrow tr.Speed_{CM} \leq tr.Speed + f(dist-to-obstacle)$

Goal Maintain [CmdedSpeedAbove7mphOfPhysicalSpeed]

FormalDef $\forall tr: Train$
 $tr.Acc_{CM} \geq 0 \Rightarrow tr.Speed_{CM} > tr.Speed + 7$

Boundary condition for logical inconsistency

$\Diamond (\exists tr: Train) (tr.Acc_{CM} \geq 0 \wedge f(dist-to-obstacle) \leq 7)$

Conflict resolution:

e.g., keep the safety goal and weaken the other

Goal Maintain [CmdedSpeedAbove7mphOfPhysicalSpeed]

FormalDef $\forall tr: Train$
 $tr.Acc_{CM} \geq 0 \Rightarrow tr.Speed_{CM} > tr.Speed + 7$
 $\vee f(dist-to-obstacle) \leq 7$

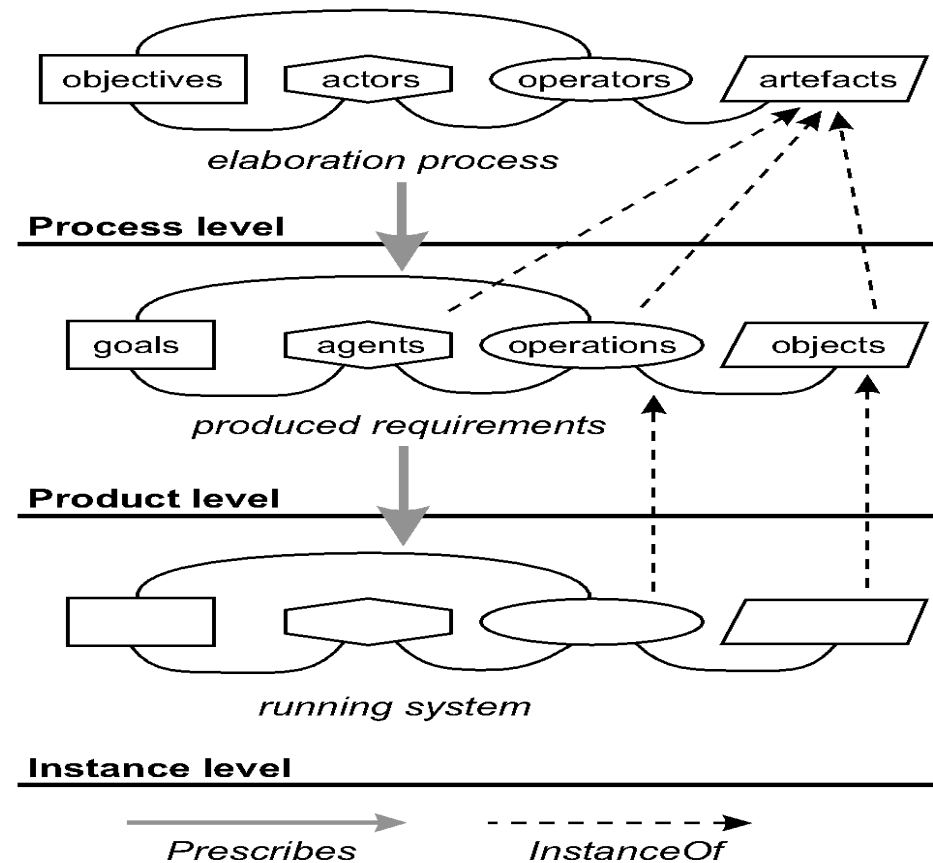
Can it get complex to have a goal-oriented reasoning?

Iteration can become a huge amount of work if the goals were not properly selected.

Stakeholders might find it difficult to understand what we are trying to accomplish. This relies in 3 points:

- the quality of the requirements definition phase
- the validation meetings during which stakeholders review consolidated goal diagrams
- refinement techniques that guarantee completeness and consistency

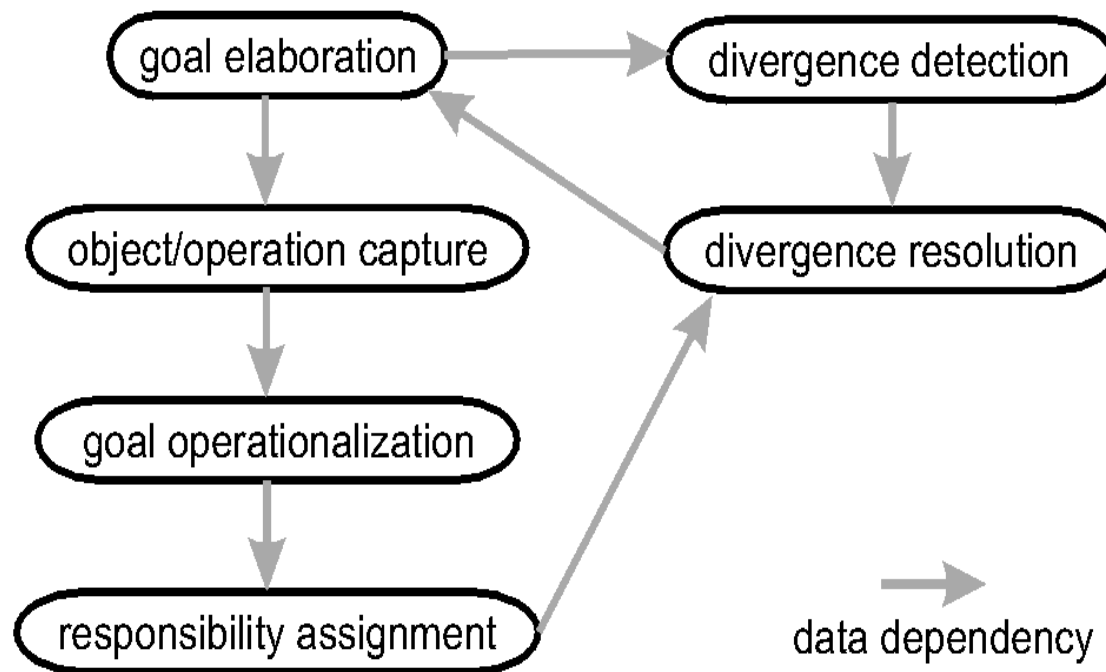
How conflicts between goals can be detected and resolved?



Classifying divergences

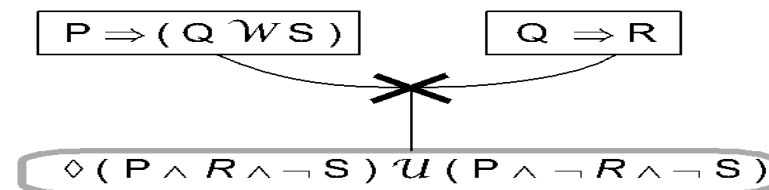
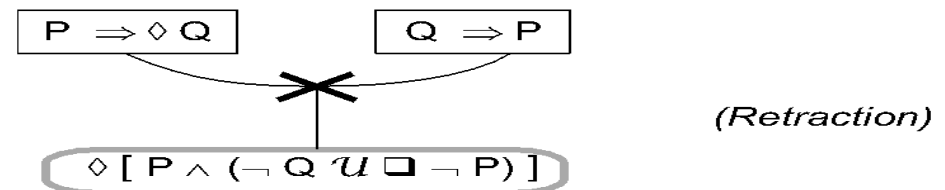
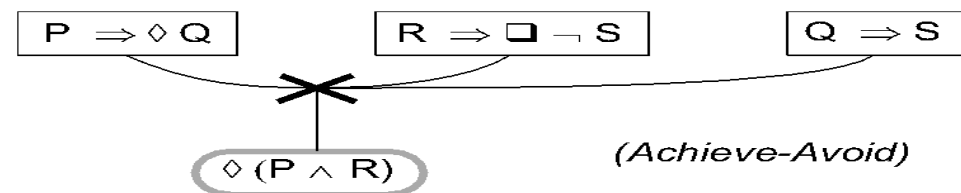
- Process-Level Deviation
- Instance-Level Deviation
- Terminology Clash
- Designation Clash
- Structure Clash
- Conflict assertions
- Competition
- Obstruction

Conflict management in goal-driven RE



Detecting divergences

- Regressing Negated Assertions
 $\{ \text{Dom}, B, \bigwedge_{j \neq i} A_j \} \vdash \neg A_i$
- Using Divergence Patterns
- Heuristics



Resolving divergences

- Assertion transformation
 - Avoiding Boundary Conditions
 - Goal Restoration
 - Conflict Anticipation
 - Goal Weakening
 - Alternative Goal Refinement
 - Divergence Resolution Heuristics
- Object transformation
 - Object Refinement
 - Agent Refinement

Questions

- Do you think that Goal-Directed Requirements Acquisition and using such methods as KAOS is a good way or Requirements Elicitation or is it too complicated?
- At which point do we iterate regarding the WHY and HOW goal gathering?

References

Francis T. Marchese, Computer science department, PACE university, 2011

Lawrence Chung, University of Texas at Dallas, 2000