Goal-Oriented Requirements Engineering (GORE)

An Introduction

Presented by: Jinwen Liu
Date: July 18, 2017
Outline

● GORE Concepts
● Advantages of GORE
● A Case Study
● Takeaway Messages
GORE Concepts - Goal

- Goals are statements of intentions and desired outcomes of a system under consideration [2][7][12].
GORE Concepts - Goal

- Goals are statements of intentions and desired outcomes of a system under consideration [2][7][12].
- Different abstraction levels
GORE Concepts - Goal

● Goals are statements of intentions and desired outcomes of a system under consideration [2][7][12].
● Different abstraction levels
  ○ Highest level: Enterprise’s wish to survive [2][8]
    ■ E.g., For a commercial enterprise, maintain good financial standing.
GORE Concepts - Goal

- Goals are statements of intentions and desired outcomes of a system under consideration [2][7][12].
- Different abstraction levels
  - Highest level: Enterprise’s wish to survive [2][8]
    - E.g., For a commercial enterprise, maintain good financial standing.
  - High level: Strategic concerns [12]
    - E.g., For a train transportation system, serve more passengers.
GORE Concepts - Goal

- Goals are statements of intentions and desired outcomes of a system under consideration [2][7][12].
- Different abstraction levels
  - Highest level: Enterprise’s wish to survive [2][8]
    - E.g., For a commercial enterprise, maintain good financial standing.
  - High level: Strategic concerns [12]
    - E.g., For a train transportation system, serve more passengers
  - Low level: Technical concerns [12]
    - E.g. For an ATM system, provide access to cash only after user removing their cards.
GORE Concepts - GORE

GORE is concerned with the use of goals in various activities of requirement engineering such as eliciting, elaborating, structuring, specifying, analyzing, negotiating, documenting, and modifying requirements[12].
GORE Concepts - Agent

- Active components in a system, such as humans, devices, and software.
- Goals are achieved by agents.
- Goal refinement process decomposes a goal into subgoals till each subgoal can be achieved by a single agent [2] [12].
GORE Concepts - Goal Types

Functional/non-functional goals:

- Functional goals: functions of the service, e.g., [12]
  - Satisfaction goals: goals that satisfy agent requests
  - Information goals: goals that provide information of objects to agents
GORE Concepts - Goal Types

Functional/non-functional goals:

- Non-functional goals: quality of the service regarding to safety, security, accuracy, performance, usability, etc. [6][12]
GORE Concepts - Goal Types

Functional/non-functional goals:

● Non-functional goals: quality of the service regarding to safety, security, accuracy, performance, usability, etc.[6][12]
  ○ Non-functional goals often receive less attention compared with functional goals in RE process, which may lead to low performances or failures of the systems [4][11].
GORE Concepts - Goal Types

Functional/non-functional goals:

- Non-functional goals: quality of the service regarding to safety, security, accuracy, performance, usability, etc. [6] [12]
  - Non-functional goals often receive less attention compared with functional goals in RE process, which may lead to low performances or failures of the systems [4] [11].
  - E.g., time/space performance goals; security goals regarding to confidentiality, integrity, availability [4] [12].
GORE Concepts - Goal Types Cont’d

Soft/hard goals:

● Soft goals: Goals that have no clear-cut criteria for achievement [8].
Soft/hard goals:

- Soft goals: Goals that have no clear-cut criteria for achievement [8].
  - Possibilities for unclear criteria:
    - Stakeholders do not agree if the goal is achieved or still ongoing.
    - Stakeholders agree the goal is achieved but don’t agree if the goal is achieved completely.
Soft/hard goals:

- Soft goals: Goals that have no clear-cut criteria for achievement [8].
  - Possibilities for unclear criterial:
    - Stakeholders do not agree if the goal is achieved or still ongoing.
    - Stakeholders agree the goal is achieved but don’t agree if the goal is achieved completely.
  - E.g., Happy customers.
GORE Concepts - Goal Types Cont’d

Soft/hard goals:

- **Hard goals:** Goals whose achievement can be verified via techniques[8].
  - E.g., keep the card if the user enter 3 wrong passwords continuously for an ATM machine
GORE Concepts - Goal Types Cont’d

Temporal behavior suggested by the goals [12]

- Achieve goals
  - Require some target properties to be satisfied in some future states.
  - E.g., Command message issued in time for a train system.
Temporal behavior suggested by the goals

- Maintain goals
  - Require some properties to be satisfied in every future state.
  - E.g., For a Mars rover, maintaining battery level of at least 20% [5]
GORE Concepts - Goal Attributes

- Name [12]
- Specification [12]
- Priority [12]
- Utility [9]
- Feasibility [9]
GORE Concepts - Goal Links

Goal links are introduced to relate goals with each other and with other elements in the requirement models. [12]
GORE Concepts - Goal Links

Links between goals

- Inter-goal contribution link
  - Links between goals that are used to capture positive/negative contribution of one goal to other goals.
GORE Concepts - Goal Links

Links between goals

- Inter-goal contribution link
  - Links between goals that are used to capture positive/negative contribution of one goal to other goals.
- AND refinement link
  - Satisfying all subgoals are sufficient for satisfying the parent goal
GORE Concepts - Goal Links

Links between goals

- Inter-goal contribution link
  - Links between goals that are used to capture positive/negative contribution of one goal to other goals.

- AND refinement link
  - Satisfying all subgoals are sufficient for satisfying the parent goal

- OR refinement link
  - Satisfying any of the subgoal is sufficient for satisfying the parent goal
GORE Concepts - Goal Links

Links between goals

- Inter-goal contribution link
  - Links between goals that are used to capture positive/negative contribution of one goal to other goals.
- AND refinement link
  - Satisfying all subgoals are sufficient for satisfying the parent goal
- OR refinement link
  - Satisfying any of the subgoal is sufficient for satisfying the parent goal
- Conflict link
  - Satisfying one goal may prevent from satisfying other goals
Links between goals and scenarios

- A scenario is a temporal sequence of interaction events between the software-to-be and its environment in the restricted context of achieving some implicit purpose(s)[5].
- Linking scenarios with goals can be helpful for requirements elicitation and validation[1].
- E.g., For a commercial system
  - Scenario: user selects “debit card” method as payment method
  - Goal: make users select (at least one) payment methods
Links between goals and agents[12]

- **Responsibility links**
  - Relate goals and agents
  - Assign goals to agents

- **Wish links**
  - An agent can wish achievement of some goals
  - We should not assign a goal to an agent that is wishing for other goals in conflict with current goal.
Advantages of GORE

● Achieving requirement completeness
  ○ If all goals from a specification can be proved to be achieved then the requirement is completed.

● Avoiding irrelevant requirements
  ○ If requirement specification can be used to achieve at least one goal then it is relevant.
Advantages of GORE

● Providing rationale for requirements
  ○ Goals provide base for requirements and make it easier to explain requirements to stakeholders.

● Providing roots for detecting and resolving requirement conflicts
  ○ Conflicts can be identified and resolved by using conflict links between goals.
Advantages of GORE

- Managing requirement evolution
  - Requirement can be viewed as one way to achieve that goal.
  - One requirement can evolve to another requirement without changing underlying goals.
  - High-level goals are more stable thus may be shared by different versions of a system.
  - Current version and future version of a system may correspond to common goals with alternative refinements and therefore can be integrated into one single goal graphs.
Case Study

The control of speed and acceleration of trains in BART (San Francisco Bay Area Rapid Transit) system [12]

- Real system
- Complex, real-time, safety-critical
- Initial document [13] (informal specification) is unbiased and publically available
Case Study Cont’d

Identifying goals from the initial document

- Search intentional keywords
  - E.g. “objective”, “goal”, “purpose”, “intent”, “in order to”, etc.
Case Study Cont’d

Identifying goals from the initial document

● Search intentional keywords
  ○ E.g. “objective”, “goal”, “purpose”, “intent”, “in order to”, etc.

● Identify soft goals
  ○ E.g. “ServeMorePassengers”, “Minimize[DevelopmentCost]”
Case Study Cont’d

A snapshot of goal graph constructed from previous step.

++ contributes strongly

-- conflicts strongly

soft goals

formalizable goals

OR refinement
Case Study Cont’d

Formalizing goals and identifying objects

Goal: Maintain [TrackSegmentSpeedLimit]

Informal Def: A train should stay below the maximum speed the track segment can handle.

Formal Def: $\forall \ tr: \text{Train}, s: \text{TrackSegment}: \text{On}(tr, s) \Rightarrow tr.\text{Speed} \leq s.\text{SpeedLimit}$
Case Study Cont’d

Eliciting new goals via WHY questions

- Higher level goals can be identified by asking WHY questions for lower level goals

- E.g., Why Maintain[WCS-DistBetweenTrains]

  => parent goal: Avoid[TrainCollision]
Eliciting new subgoals via HOW questions

- More concrete goals can be identified by asking HOW questions
- E.g., How: Maintain [WCS-DistBetweenTrains]
Identifying responsibility assignment

- Subgoals are refined via HOW questions in the previous step until each subgoal can be assigned to individual agents.

- Examples of responsibility assignment
  - Maintain[AccurateSpeed/PositionEstimates] => TrackingSystem agent
  - Maintain[SafeTrainResponseToCommand] => OnBoardTrainController agent
Case Study Cont’d

Handling conflicts

- Train commanded speed should not be too high => safe transport
- Train commanded speed should not be too low => smooth move
Case Study Cont’d

**Before**

**Goal**: Maintain \([\text{CmdedSpeedCloseToPhysicalSpeed}]\)

**FormalDef**: \( \forall \text{tr: Train} \)

\[ \text{tr.Acc}_{CM} \geq 0 \]

\[ \Rightarrow \text{tr.Speed}_{CM} \leq \text{tr.Speed} + f(\text{dist-to-obstacle}) \]

**Goal**: Maintain \([\text{CmdedSpeedAbove7mphOfPhysicalSpeed}]\)

**FormalDef**: \( \forall \text{tr: Train} \)

\[ \text{tr.Acc}_{CM} \geq 0 \Rightarrow \text{tr.Speed}_{CM} > \text{tr.Speed} + 7 \]
Case Study Cont’d

Before

**Goal** Maintain \([\text{CmdedSpeedCloseToPhysicalSpeed}]\)

**FormalDef** \( \forall \ tr: \text{Train} \)
\[
\begin{align*}
\text{tr.Acc}_{CM} & \geq 0 \\
\Rightarrow \text{tr.Speed}_{CM} & \leq \text{tr.Speed} + f(\text{dist-to-obstacle})
\end{align*}
\]

**Goal** Maintain \([\text{CmdedSpeedAbove7mphOfPhysicalSpeed}]\)

**FormalDef** \( \forall \ tr: \text{Train} \)
\[
\begin{align*}
\text{tr.Acc}_{CM} & \geq 0 \\
\Rightarrow \text{tr.Speed}_{CM} & > \text{tr.Speed} + 7
\end{align*}
\]

Conflict if \( \text{tr.Acc}_{CM} \geq 0 \land f(\text{dist-to-obstacle}) \leq 7 \)
Case Study Cont’d

Before

Goal Maintain [CmdedSpeedCloseToPhysicalSpeed]

FormalDef \( \forall \text{tr: Train} \)

\[ \text{tr.Acc}_{CM} \geq 0 \]

\[ \Rightarrow \text{tr.Speed}_{CM} \leq \text{tr.Speed} + f(\text{dist-to-obstacle}) \]

Goal Maintain [CmdedSpeedAbove7mphOfPhysicalSpeed]

FormalDef \( \forall \text{tr: Train} \)

\[ \text{tr.Acc}_{CM} \geq 0 \Rightarrow \text{tr.Speed}_{CM} > \text{tr.Speed} + 7 \]

After

Goal Maintain [CmdedSpeedCloseToPhysicalSpeed]

FormalDef \( \forall \text{tr: Train} \)

\[ \text{tr.Acc}_{CM} \geq 0 \]

\[ \Rightarrow \text{tr.Speed}_{CM} \leq \text{tr.Speed} + f(\text{dist-to-obstacle}) \]

Goal Maintain [CmdedSpeedAbove7mphOfPhysicalSpeed]

FormalDef \( \forall \text{tr: Train} \)

\[ \text{tr.Acc}_{CM} \geq 0 \Rightarrow \text{tr.Speed}_{CM} > \text{tr.Speed} + 7 \]

\[ \vee f(\text{dist-to-obstacle}) \leq 7 \]

Conflict if \( \text{tr.Acc}_{CM} \geq 0 \land f(\text{dist-to-obstacle}) \leq 7 \)
Takeaway Messages

- GORE is concerned with the use of goals in various activities of requirement engineering (e.g., requirement elicitation, analysis, specification, validation, etc.)
Takeaway Messages

- GORE is concerned with the use of goals in various activities of requirement engineering (e.g., requirement elicitation, analysis, specification, validation, etc.)
- GORE has many advantages (e.g., proving completeness and rationale for requirements, detecting and managing conflicts, etc.)
Takeaway Messages

● GORE is concerned with the use of goals in various activities of requirement engineering (e.g., requirement elicitation, analysis, specification, validation, etc.)
● GORE has many advantages (e.g., proving completeness and rationale for requirements, detecting and managing conflicts, etc.)
● GORE is widely used in practice and studied in research.
Thank you!

Q&A


References


