Tropos: Goal Analysis

Paolo Giorgini
Department of Information and Communication Technology
University of Trento - Italy
http://www.dit.unitn.it/~pgiorgio

Agent-Oriented Software Engineering course
Laurea Specialistica in Informatica
A.A. 2009-2010
Outline

- The concept of goal in SE
  - Goal analysis
- Tropos Goal Reasoning Framework
  - Qualitative approach
  - Quantitative approach
  - Backward and forward reasoning
  - GR-Tool
Goals

- The concept of goal has been used in many area of Computer Science, e.g.,
  - in planning to describe desirable states of the world
  - in agent architecture to describe agents’ mental state

- More recently, goals have been used in Software Engineering to model:
  - Early requirements (e.g., *every book request will eventually be fulfilled*)
  - Non-functional requirements (e.g., *the new system will be highly reliable*)
Goal Analysis

- Traditionally, goal analysis consists of decomposing goals into subgoals through an AND- or OR-decomposition.

- Given a goal model and a set of initial labels for some goals (S for Satisfied and D for Denied) there is a simple labels propagation algorithm which can generate labels for all other goals of the model [Nilsson’72]
Goal Analysis

- However, there are many domains where goals are not formalizable and the relationships among them cannot be captured by semantically well-defined relations such as AND/OR ones.

- E.g., in RE:
  - Highly reliable system, has no formally defined predicate to prescribe its meaning, though you can define necessary conditions for its satisfaction
  - Highly reliable system can be related to other goals, such as thoroughly debagged system (the latter contributes to the satisfaction of the former) - partial and qualitative contribution
A simple example of goal model

- AND/OR decompositions
- Positive (+)/ Negative (-) contribution links
Part of a goal model for General Motor
Goal modeling and reasoning

- In Tropos we need to capture relations among goals/softgoals
  - From early requirements analysis up to architectural design
- We need also to reason about satisfaction or denial of goals
  - If all subgoals are satisfied, top goals are satisfied
  - But what happen if a subgoal is partially satisfied?
  - … and what happen if two goals are in conflict (E.g., prepare the AOSE course and go to the beach)?
- Different form of reasoning
  - What the minimal set of subgoals that allow me to satisfy all top goals?
  - If I satisfy a specific subset of leaf goals what happen to my top goals?
  - Qualitative reasoning (E.g., a goal is partially satisfied)
  - Quantitative reasoning (E.g, the probability for a goal to be satisfied)
The Tropos approach

- Evidence about satisfaction/denial of goals
- Reasoning mechanisms to propagate evidence in the model
- The reasoning output is used to support the analysis and design process
Goal Models

- **Goal Dependency Graph:**
  - Goals represented as Nodes
  - And/or relationships as (grouped) and/or arcs
  - Identity/negation as ++/- - arcs
  - Positive/negative contribution as +/- arcs
  - Cycles possible!

- **Goal Valuations:**
  - Goals can be either satisfied or denied
    - need to represent evidence of satisfaction/denial
  - Relationships propagate satisfaction and denial values
  - Conflicts possible!
The problem

Provide:

- Formal representation(s) of goal models and goal valuations
  - Qualitative and quantitative approach
- Formal techniques to reason on goal values and on their propagation through goal models
  - Top-down (backward) reasoning
  - Bottom-up (forward) reasoning
Qualitative approach

- Four predicates:
  - FS(g): there is at least Full evidence that g is Satisfied
  - PS(g): there is at least Partial evidence that g is Satisfied
  - FD(g): there is at least Full evidence that g is Denied
  - PD(g): there is at least Partial evidence that g is Denied

- Negated atoms $\neg$FS(g), $\neg$FD(g) not admitted!
- FS(g)/PS(g) independent from FD(g)/PD(g)
  - This allow to have conflicts
    - E.g., $g$ can be fully satisfied and partially denied
    - Different sources of information can provide both evidence for satisfaction and denial
    - A goal can receive a negative contribution from another goal (you cannot do both!) but its actual decomposition allow to satisfy the goal
Axiomatization

- Axioms allow to capture (define) the semantics of goal models
  - Express the semantics of relations and value propagation
  - Used to build sound reasoning techniques

<table>
<thead>
<tr>
<th>Goal</th>
<th>Invariant Axioms</th>
</tr>
</thead>
<tbody>
<tr>
<td>g</td>
<td>FS(g) → PS(g)</td>
</tr>
<tr>
<td></td>
<td>FD(g) → PD(g)</td>
</tr>
</tbody>
</table>
Axiomatization

<table>
<thead>
<tr>
<th>Goal Relation</th>
<th>Relation Axioms</th>
</tr>
</thead>
</table>
| (G2, G3) \(\rightarrow\) G1: AND  | \[(FS(G2) \land FS(G3)) \rightarrow FS(G1)\]  
|              | \[(PS(G2) \land PS(G3)) \rightarrow PS(G1)\]  
|              | FD(G2) \rightarrow FD(G1), FD(G3) \rightarrow FD(G1)  
|              | PD(G2) \rightarrow PD(G1), PD(G3) \rightarrow PD(G1)  |
| (G2, G3) \(\rightarrow\) G1: OR   | \[(FS(G2) \lor FS(G3)) \rightarrow FS(G1)\]  
|              | \[(PS(G2) \lor PS(G3)) \rightarrow PS(G1)\]  
|              | FD(G2) \rightarrow FD(G1), FD(G3) \rightarrow FD(G1)  
|              | PD(G2) \rightarrow PD(G1), PD(G3) \rightarrow PD(G1)  |
Axiomatization

<table>
<thead>
<tr>
<th>Goal Relation</th>
<th>Relation Axioms</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_2 \rightarrow G_1$: $^{++S}$</td>
<td>$\text{FS}(G_2) \rightarrow \text{FS}(G_1)$, $\text{PS}(G_2) \rightarrow \text{PS}(G_1)$</td>
</tr>
<tr>
<td>$G_2 \rightarrow G_1$: $^{--S}$</td>
<td>$\text{FS}(G_2) \rightarrow \text{FD}(G_1)$, $\text{PS}(G_2) \rightarrow \text{PD}(G_1)$</td>
</tr>
<tr>
<td>$G_2 \rightarrow G_1$: $^{+S}$</td>
<td>$\text{FS}(G_2) \rightarrow \text{PS}(G_1)$, $\text{PS}(G_2) \rightarrow \text{PS}(G_1)$</td>
</tr>
<tr>
<td>$G_2 \rightarrow G_1$: $^{-S}$</td>
<td>$\text{FS}(G_2) \rightarrow \text{PD}(G_1)$, $\text{PS}(G_2) \rightarrow \text{PD}(G_1)$</td>
</tr>
<tr>
<td>$G_2 \rightarrow G_1$: $^{++D}$</td>
<td>$\text{FD}(G_2) \rightarrow \text{FD}(G_1)$, $\text{PD}(G_2) \rightarrow \text{PD}(G_1)$</td>
</tr>
<tr>
<td>$G_2 \rightarrow G_1$: $^{--D}$</td>
<td>$\text{FD}(G_2) \rightarrow \text{FS}(G_1)$, $\text{PD}(G_2) \rightarrow \text{PS}(G_1)$</td>
</tr>
<tr>
<td>$G_2 \rightarrow G_1$: $^{+D}$</td>
<td>$\text{FD}(G_2) \rightarrow \text{PD}(G_1)$, $\text{PD}(G_2) \rightarrow \text{PD}(G_1)$</td>
</tr>
<tr>
<td>$G_2 \rightarrow G_1$: $^{-D}$</td>
<td>$\text{FD}(G_2) \rightarrow \text{PS}(G_1)$, $\text{PD}(G_2) \rightarrow \text{PS}(G_1)$</td>
</tr>
</tbody>
</table>
Axiomatization (cont.)

- or, +D, -D, ++D, --D are dual w.r.t. and, +S, -S, ++S, --S
- Propagation of satisfaction through a ++, --, +, - may be or may be not symmetric w.r.t. that of denial:

\[ G_2 \rightarrow^+ G_1 \iff G_2 \rightarrow^{+S} G_1 \text{ and } G_2 \rightarrow^{+D} G_1 \]
\[ G_2 \rightarrow^- G_1 \iff G_2 \rightarrow^{-S} G_1 \text{ and } G_2 \rightarrow^{-D} G_1 \]

Satisfaction/Denial:

g is totally satisfied [resp. partially satisfied, totally/partially denied] iff

FS(g) [resp. PS(g), FD(g), PD(g)] can be logically inferred from the initial assignment and the axioms
Forward Reasoning

- **Given**
  - goal model
  - initial values assignment to some goals
    (input goals -- typically leaf goals)
  - **Forward reasoning** focuses on the forward propagation of these initial values to all other goals of the graph accordingly to the axioms
  - Initial values represents the evidence (possibly contradictory) available about the satisfaction and the denial of goals: \{FS(G1), PD(G2), \ldots\}
    - Usually provided by the domain expert(s)
An example of Forward prop.

Initial:  \{\text{FS(}teach\ the\ course\text{)},\ \text{FS(}with\ slides\text{)},\ \text{FS(}go\ to\ the\ beach\text{)}\}\n
Final:  \{\text{FS(}teach\ the\ course\text{)},\ \text{FS(}with\ slides\text{)},\ \text{FS(}go\ to\ the\ beach\text{)},\ \text{FS(}prepare\ the\ course\text{)},\ \text{PD(}prepare\ the\ course\text{)},\ \text{PS(}enjoy\ Malaga\text{)},\ \text{PS(}satisfy\ students\ needs\text{)},\ \text{PS(}teach\ high\ quality\ course\text{)})\ldots\}\
Propagation Algorithm

1. \textit{label\_array} \textit{Label\_Graph}(graph \langle G, R \rangle, \textit{label\_array Initail})
2. \quad \textit{Current}=\textit{Initial};
3. \quad \textbf{do}
4. \quad \quad \textit{Old}=\textit{Current};
5. \quad \quad \textbf{for each } \textit{Gi} \in \textit{G} \textbf{ do}
6. \quad \quad \quad \textit{Current}[i]=\textit{Update\_label}(i, \langle G, R \rangle, \textit{Old});
7. \quad \quad \textbf{until not (} \textit{Current}==\textit{Old} \textbf{);} 
8. \quad \textbf{return } \textit{Current};

17. \textit{label} \textit{Update\_label}(\textit{int} \textit{i}, \textit{graph} \langle \textit{G}, \textit{R} \rangle, \textit{label\_array Old})
18. \quad \textbf{for each } \textit{Rj} \in \textit{R} \textbf{ s.t. target(Ri)== Gi \textbf{ do}}
19. \quad \quad \textit{satij} = \textit{Apply\_Rules\_Sat}(\textit{Gi}, \textit{Rj}, \textit{Old})
20. \quad \quad \textit{denij} = \textit{Apply\_Rules\_Den}(\textit{Gi}, \textit{Rj}, \textit{Old})
21. \quad \textbf{return} \langle \textit{max}(\textit{maxj}(\textit{satij}), \textit{Old}[i].sat),
\textit{max}(\textit{maxj}(\textit{denij}), \textit{Old}[i].den) \rangle
Propagation Algorithm (cont.)

- To each $g$ we associate two variables $\text{Sat}(g)$ and $\text{Den}(g)$ ranging in $\{F,P,N\}$ such that $F>P>N$
- E.g., $\text{Sat}(g) \geq P$ states that there is at least partial evidence that $g$ is satisfiable
- From the initial assignment, we propagate the values according to the following rules:

  - or, $+D$, $-D$, $++D$, $--D$ dual w.r.t. and, $+S$, $-S$, $++S$, $--S$
  - Satisfaction/denial values monotonically non-decreasing
  - Terminates when reaches a fixpoint ($\text{Current}==\text{Old}$)
<table>
<thead>
<tr>
<th>Goals</th>
<th>Exp 1</th>
<th>Exp 2</th>
<th>Exp 3</th>
<th>Exp 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Init</td>
<td>Fin</td>
<td>Init</td>
<td>Fin</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>SD</td>
<td>SD</td>
<td>SD</td>
</tr>
<tr>
<td>DB querying</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>catalogue consulting</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pick available item</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>pre-order non available item</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>classic communication handled</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>standard form order</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>secure form order</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>manage internet shop</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>privacy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>availability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>integrity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>usability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>adaptability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>easy to use</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>security</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Forward Reasoning in Tropos

- It is adopted for evaluating the impact of the adoption of the different alternatives with respect to the
  - functional requirements (top goals)
  - non-functional requirements (softgoals)

of the system-to-be

- Reasoning may involve
  - Single actor (intra actor reasoning)
  - Multiple actors (inter actor reasoning)
Backward reasoning

- We set the desired final values of the target goals, and we want to find possible initial assignments to the input goals which would cause the desired final values of the target goals.
- We search for possible initial assignments to the input goals which would cause the desired final values of the target goals by forward propagation.
- The user may also add some desired constraints, and decide to avoid conflicts.
An example of Backward prop.

Final:  \{FS(\text{give AOSE course}), \text{PS(teach high quality course)}\}
Assign. :  \{FS(\text{prepare the course}), \text{PS(with slides)}\}
Constraints and Costs

- We may also add some desired constraints and decide to avoid
  - Strong conflict (e.g., FS(G),FD(G))
  - Medium conflict (e.g., FS(G),PD(G))
  - Weak (e.g., PS(G),PD(G))
  - all conflicts

- Assigning a cost to each input goal, we search for an assignment at the minimum cost
Propositional Satisfiability (SAT)

- We reduce the backward search to a SAT problem.
- SAT is the problem of determining whether a boolean formula $\Phi$ admits at least one satisfying truth assignment $\mu$ to its variables $A_i$.
- SAT is a NP-complete problem (there does not exist any polynomial algorithm able to solve it).
- There exists efficient SAT techniques:
  - DPLL is the most popular SAT algorithm.
  - CHAFF is the most efficient DPLL implementation.
- There are several techniques to improve the efficiency of DPLL (e.g., backjumping, learning, random restart).
Minimum-Weight SAT (MW-SAT)

- MW-SAT is a variant of SAT, where the boolean variables $A_i$ occurring in $\Phi$ are given a positive integer weight $w_i$
- MW-SAT is the problem of determining a truth assignment $\mu$ satisfying $\Phi$ which minimizes the value

$$W(\mu) := \sum_i \{ w_i \mid A_i \text{ is assigned } \top \text{ by } \mu \}$$

or stating there is none.
- MINWEIGHT is the state-of-art solver for MW-SAT
Basic Formalization

- The boolean variables of $\Phi$ are all the values $FS(G), PS(G), FD(G), PD(G)$ for each goal $G$ and $\Phi$ is

$$\Phi := \Phi_{\text{graph}} \land \Phi_{\text{outval}} \land \Phi_{\text{backward}} [\land \Phi_{\text{constraints}} \land \Phi_{\text{conflict}}]$$

where

- $\Phi_{\text{graph}}$ encodes the goal graph
- $\Phi_{\text{outval}}$ encodes the desired final output values
- $\Phi_{\text{backward}}$ encodes the backward reasoning
- $\Phi_{\text{constraints}}$ encodes user’s constraints (optional)
- $\Phi_{\text{conflict}}$ encodes the prevention of conflicts (optional)
Basic Formalization cont.

- **Encoding the goal graph**

\[
\Phi_{\text{graph}} := \bigwedge_{G \in \mathcal{G}} \text{Invar}_\text{Ax}(G) \land \bigwedge_{r \in \mathcal{R}} \text{Rel}_\text{Ax}(r).
\]

*Invar\_Ax(G)* is the conjunction of the invariant axioms and *Rel\_Ax(r)* is the conjunction of the relation axioms (forward propagation through the relation arcs in the graph).

- **Representing Desired Final Output Values**

\[
\Phi_{\text{outval}} := \bigwedge_{G \in \text{Target}(\mathcal{G})} vs(G) \land \bigwedge_{G \in \text{Target}(\mathcal{G})} vd(G).
\]

*Target(\mathcal{G})* is the set of target goals and *vs(G) \in \{T,PS(G),FS(G)\}*, *vd(G) \in \{T,PD(G),FD(G)\}* are the maximum satisfiability and deniability values assigned to the target goal G.
Basic Formalization cont.

- **Econding Backward Reasoning**

\[ \Phi_{\text{backward}} := \bigwedge_{G \in \text{Input}(\mathcal{G})} \bigwedge_{v(G)} \text{Backward}_{\text{Ax}}(v(G)) \]

\[ \text{Backward}_{\text{Ax}}(v(G)) := v(G) \rightarrow \bigvee_{r \in \text{Incoming}(G)} \text{Prereqs}(v(G), r) \]

\text{Input}(\mathcal{G}) \text{ is the set of input goals; } \text{Incoming}(G) \text{ is the set of relations in } G;\]
\[ v(G) = \{PS(G), FS(G), PD(G), FD(G)\}, \text{ and } \text{Prereqs}(v(G), r) \text{ is a formula which is true iff the prerequisites of } v(G) \text{ through } r \text{ hold.} \]

\text{Backward}_{\text{Ax}}(v(G)) \text{ is the set of propagation axioms (see next slide)}

If G is not an input goal and v(G) holds, then this value must derive from the prerequisite values of some incoming relations of G.
Axioms for backward propagation

\[
\begin{align*}
FS(G) & \rightarrow \\
& \left( \bigwedge_i FS(G_i) \lor \text{If } (G_1, \ldots, G_i, \ldots, G_n) \rightarrow G \right) \\
& \lor \left( \bigvee_i FS(G_i) \lor \text{If } (G_1, \ldots, G_i, \ldots, G_n) \rightarrow G \right) \\
& \lor \left( \forall R_i : G_i \xrightarrow{S} G \right) \\
& \lor \left( \forall R_i : G_i \xrightarrow{D} G \right)
\end{align*}
\]

\[
\begin{align*}
FD(G) & \rightarrow \\
& \left( \bigwedge_i FD(G_i) \lor \text{If } (G_1, \ldots, G_i, \ldots, G_n) \rightarrow G \right) \\
& \lor \left( \bigvee_i FD(G_i) \lor \text{If } (G_1, \ldots, G_i, \ldots, G_n) \rightarrow G \right) \\
& \lor \left( \forall R_i : G_i \xrightarrow{S} G \right) \\
& \lor \left( \forall R_i : G_i \xrightarrow{D} G \right)
\end{align*}
\]

\[
\begin{align*}
PS(G) & \rightarrow \\
& \left( \bigwedge_i PS(G_i) \lor \text{If } (G_1, \ldots, G_i, \ldots, G_n) \rightarrow G \right) \\
& \lor \left( \bigvee_i PS(G_i) \lor \text{If } (G_1, \ldots, G_i, \ldots, G_n) \rightarrow G \right) \\
& \lor \left( \forall R_i : G_i \xrightarrow{S} G \right) \\
& \lor \left( \forall R_i : G_i \xrightarrow{D} G \right)
\end{align*}
\]

\[
\begin{align*}
PD(G) & \rightarrow \\
& \left( \bigwedge_i PD(G_i) \lor \text{If } (G_1, \ldots, G_i, \ldots, G_n) \rightarrow G \right) \\
& \lor \left( \bigvee_i PD(G_i) \lor \text{If } (G_1, \ldots, G_i, \ldots, G_n) \rightarrow G \right) \\
& \lor \left( \forall R_i : G_i \xrightarrow{S} G \right) \\
& \lor \left( \forall R_i : G_i \xrightarrow{D} G \right) \\
& \lor \left( \forall R_i : G_i \xrightarrow{S} G \right) \\
& \lor \left( \forall R_i : G_i \xrightarrow{D} G \right)
\end{align*}
\]

© P. Giorgini
Optional components

Adding User’s Constraints and Desiderata

The user expresses constraints and desiderata on goal values (e.g., “PS(G₁)” means “G₁ is at least partial satisfiable”, but it might totally satisfiable.

A negative clause value is used to prevent a value to a goal (e.g., “¬FD(G₁)” means “G₁ cannot be fully deniable”, but it might be partially deniable).

FS(G₁) ∨ FS(G₂) means at least G₁ or G₂ must be fully satisfiable.
Basic Formalization cont.

Preventing conflicts

It allows the user for looking for solutions which do not involve conflicts

Strong conflicts

\[ \Phi_{conflict} := \bigwedge_{G \in \mathcal{G}} (\neg F S(G) \lor \neg F D(G)) \]

Strong and medium conflicts

\[ \Phi_{conflict} := \bigwedge_{G \in \mathcal{G}} ((\neg F S(G) \lor \neg P D(G)) \land (\neg P S(G) \lor \neg F D(G)) \]

All conflicts

\[ \Phi_{conflict} := \bigwedge_{G \in \mathcal{G}} (\neg P S(G) \lor \neg P D(G)) \]
Backward Prop. implementation

- We have reduced the qualitative problem to the Satisfiability (SAT) and minimum-cost satisfiability (minimum-cost SAT) problems for Boolean formulas
- GOLSOLVE / GOLMINSOLVE
Backward reasoning in Tropos

- Used to find the set of goals at the minimum costs that if achieved they can guarantee the achievement of the desired top goals (functional requirements) and softgoals (non-functional requirements).
- In other words, we find among the alternatives of the goal model those with the minimal cost that allow us to obtain our desired goals.
- Reasoning may involve
  - Single actor (intra actor reasoning)
  - Multiple actors (inter actor reasoning)
<table>
<thead>
<tr>
<th>Goals</th>
<th>Exp 1</th>
<th>Exp 2</th>
<th>Exp 3</th>
<th>Exp 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Init</td>
<td>Fin</td>
<td>Init</td>
<td>Fin</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>SD</td>
<td>SD</td>
<td>SD</td>
</tr>
<tr>
<td>DB querying (3)</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>catalogue consulting (6)</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>pick available item (2)</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>pre-order non available item (7)</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>classic communication handled (4)</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>standard form order (6)</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>secure form order (8)</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>manage internet shop</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>privacy</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>availability</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>integrity</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>usability</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>adaptability</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>easy to use</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>security</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

Avoiding conflicts
Quantitative approach
Quantitative Approach

- Evidence of satisfaction/denial represented by real values in \( D : [\inf, \sup], 0 \leq \inf < \sup \)
- Value propagation through goal graphs as math functions, \( f : D^n \rightarrow D \)
- Much finer-grained:
  - Different degrees of satisfaction/denial evidence
  - Different degrees of positive/negative contribution
  - Different strength of conflicts
Numerical representation of evidence

- $\text{Sat}(g), \text{Den}(g) \in [\text{inf}, \text{sup}]$
- **Atoms in the form** $\text{Sat}(g) \geq c_1 [\text{Den}(g) \geq c_2]$: “there is at least an evidence $c_1 [c_2]$ that $g$ is Satisfied [Denied]”
  
  \[
  c_1 = \text{inf}, \ c_2 = \text{inf} \iff \top
  \]
  
  \[
  c_1, c_2 \in ]\text{inf}, \text{sup}[ \iff \text{PS}(g), \text{PD}(g)
  \]
  
  \[
  c_1 = \text{sup}, c_2 = \text{sup} \iff \text{FS}(g), \text{FD}(g)
  \]
- **Conflict**: $\text{Sat}(g) \geq c_1$ and $\text{Den}(g) \geq c_2$, $c_1, c_2 \in ]\text{inf}, \text{sup}[$
Value propagation model

- 2 dual OPERATORS: $\oplus$ and $\otimes$, representing value propagation through “or” and “and”
  - Independent probability model:
    \begin{align*}
    \text{inf} &= 0, \text{sup} = 1 \\
    p_1 \oplus p_2 &= p_1 + p_2 - p_1 \cdot p_2 \\
    p_1 \otimes p_2 &= p_1 \cdot p_2
    \end{align*}
  - Flow model (Resistor):
    \begin{align*}
    \text{inf} &= 0, \text{sup} = +\infty \\
    v_1 \oplus v_2 &= v_1 + v_2 \\
    v_1 \otimes v_2 &= (v_1 \cdot v_2)/(v_1 + v_2)
    \end{align*}
- ...
Axiomatization

<table>
<thead>
<tr>
<th>Goal Relation</th>
<th>Relation Axioms</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AND</strong></td>
<td></td>
</tr>
<tr>
<td>$(G_2, G_3) \rightarrow G_1$:</td>
<td>$(\text{Sat}(G_2) \geq x \land \text{Sat}(G_3) \geq y) \rightarrow \text{Sat}(G_1) \geq (x \otimes y)$</td>
</tr>
<tr>
<td></td>
<td>$(\text{Den}(G_2) \geq x \land \text{Den}(G_3) \geq y) \rightarrow \text{Den}(G_1) \geq (x \oplus y)$</td>
</tr>
<tr>
<td><strong>OR</strong></td>
<td></td>
</tr>
<tr>
<td>$(G_2, G_3) \rightarrow G_1$:</td>
<td>$(\text{Sat}(G_2) \geq x \lor \text{Sat}(G_3) \geq y) \rightarrow \text{Sat}(G_1) \geq (x \oplus y)$</td>
</tr>
<tr>
<td></td>
<td>$(\text{Den}(G_2) \geq x \lor \text{Den}(G_3) \geq y) \rightarrow \text{Den}(G_1) \geq (x \otimes y)$</td>
</tr>
</tbody>
</table>

- AND and OR relation are dual
## Axiomatization cont.

<table>
<thead>
<tr>
<th>Goal Relation</th>
<th>Relation Axioms</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ G_2 \rightarrow G_1: ] [ w^+ S ]</td>
<td>[ \text{Sat}(G_2) \geq x \rightarrow \text{Sat}(G_1) \geq (x \otimes w) ]</td>
</tr>
<tr>
<td>[ G_2 \rightarrow G_1: ] [ w^- S ]</td>
<td>[ \text{Sat}(G_2) \geq x \rightarrow \text{Den}(G_1) \geq (x \otimes w) ]</td>
</tr>
<tr>
<td>[ G_2 \rightarrow G_1: ] [ ++ S ]</td>
<td>[ \text{Sat}(G_2) \geq x \rightarrow \text{Sat}(G_1) \geq x ]</td>
</tr>
<tr>
<td>[ G_2 \rightarrow G_1: ] [ -- S ]</td>
<td>[ \text{Sat}(G_2) \geq x \rightarrow \text{Den}(G_1) \geq x ]</td>
</tr>
<tr>
<td>[ G_2 \rightarrow G_1: ] [ w^+ D ]</td>
<td>[ \text{Den}(G_2) \geq x \rightarrow \text{Den}(G_1) \geq (x \otimes w) ]</td>
</tr>
<tr>
<td>[ G_2 \rightarrow G_1: ] [ w^- D ]</td>
<td>[ \text{Den}(G_2) \geq x \rightarrow \text{Sat}(G_1) \geq (x \otimes w) ]</td>
</tr>
<tr>
<td>[ G_2 \rightarrow G_1: ] [ ++ D ]</td>
<td>[ \text{Den}(G_2) \geq x \rightarrow \text{Den}(G_1) \geq x ]</td>
</tr>
<tr>
<td>[ G_2 \rightarrow G_1: ] [ -- D ]</td>
<td>[ \text{Den}(G_2) \geq x \rightarrow \text{sat}(G_1) \geq x ]</td>
</tr>
</tbody>
</table>

- +D, -D, ++D, --D dual w.r.t. +S, -S, ++S, --S
- Remark: + and - relations have a weight \( w \)
Quantitative propagation

- There is at least an evidence $c$ that $g$ is satisfied [resp. denied] iff $\text{Sat}(g) \geq c$ [resp. $\text{Den}(g) \geq c$] can be logically inferred from the initial assignment and the axioms.

- $\text{Sat}(g) \geq c, \text{Den}(g) \geq c$ propagated independently
Forward Propagation Algorithm

1. \textit{label\_array Label\_Graph(graph }\langle\mathbf{G}, \mathbf{R}\rangle, \textit{label\_array Initial})
2. \textit{Current=Initial;}
3. \textit{do}
4. \textit{Old=Current;}
5. \textit{for each } \mathbf{G}_i \in \mathbf{G} \textit{ do}
6. \hspace{1em} \textit{Current}[i]=\textit{Update\_label}(i, \langle\mathbf{G}, \mathbf{R}\rangle, \textit{Old});
7. \textit{until not } (\|\textit{Current} – \textit{Old}\|_\infty \leq \varepsilon);
8. \textit{return } \textit{Current;}

17. \textit{label Update\_label(int } i \textit{, graph }\langle\mathbf{G}, \mathbf{R}\rangle, \textit{label\_array Old})
18. \textit{for each } \mathbf{R}_j \in \mathbf{R} \textit{ s.t. } \textit{target}(\mathbf{R}_i) == \mathbf{G}_i \textit{ do}
19. \hspace{1em} \textit{sat}_{ij} = \textit{Apply\_Rules\_Sat}(\mathbf{G}_i, \mathbf{R}_j, \textit{Old})
20. \hspace{1em} \textit{den}_{ij} = \textit{Apply\_Rules\_Den}(\mathbf{G}_i, \mathbf{R}_j, \textit{Old})
21. \textit{return } \langle \max(\max_j(\textit{sat}_{ij}), \textit{Old}[i].\textit{sat}), \max(\max_j(\textit{den}_{ij}), \textit{Old}[i].\textit{den}) \rangle
Forward Propagation Algorithm

- Satisfaction/denial values monotonically non-decreasing
- Uses Cauchy-convergence as termination condition:
  \[ |a_{n+1} - a_n| \xrightarrow{n \to \infty} 0 \]
Quantitative approach: example

<table>
<thead>
<tr>
<th>Goal/Event</th>
<th>Relationship</th>
<th>Goal/Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>increase sales volume</td>
<td>0.6 _S</td>
<td>increase Toyota sales</td>
</tr>
<tr>
<td>increase Toyota sales</td>
<td>0.6 _S</td>
<td>increase VW sales</td>
</tr>
<tr>
<td>increase VW sales</td>
<td>0.6 _S</td>
<td>increase sales volume</td>
</tr>
<tr>
<td>increase customer loyalty</td>
<td>0.4 _S</td>
<td>increase sales volume</td>
</tr>
<tr>
<td>increase sales prices</td>
<td>0.5 _S</td>
<td>increase customer loyalty</td>
</tr>
<tr>
<td>increase car quality</td>
<td>0.8 _S</td>
<td>increase customer loyalty</td>
</tr>
<tr>
<td>improve car services</td>
<td>0.7 _S</td>
<td>increase customer loyalty</td>
</tr>
<tr>
<td>lower environment impact</td>
<td>0.4 _S</td>
<td>increase customer loyalty</td>
</tr>
<tr>
<td>increase sales prices</td>
<td>0.8 _S</td>
<td>improve car services</td>
</tr>
<tr>
<td>keep labour costs low</td>
<td>0.7 _S</td>
<td>increase car quality</td>
</tr>
<tr>
<td>improve economies of production</td>
<td>0.8 _S</td>
<td>lower purchase costs</td>
</tr>
<tr>
<td>Yen rises</td>
<td>0.8 _S</td>
<td>increase foreign earnings</td>
</tr>
<tr>
<td>lower Japanese interest rates</td>
<td>0.4 _S</td>
<td>lower sales price</td>
</tr>
<tr>
<td>Japanese rates rises</td>
<td>0.6 _S</td>
<td>lower Japanese interest rates</td>
</tr>
<tr>
<td>Japanese rates rises</td>
<td>0.4 _S</td>
<td>Yen rises</td>
</tr>
<tr>
<td>Yen rises</td>
<td>0.4 _S</td>
<td>Japanese gas price rises</td>
</tr>
<tr>
<td>Japanese gas price rises</td>
<td>0.6 _S</td>
<td>gas price rises</td>
</tr>
<tr>
<td>US gas price rises</td>
<td>0.6 _S</td>
<td>gas price rises</td>
</tr>
<tr>
<td>gas price rises</td>
<td>0.4 _S</td>
<td>improve mileage</td>
</tr>
<tr>
<td>Goals/Events</td>
<td>Exp 1</td>
<td></td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td>Init</td>
<td>Fin</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>D</td>
</tr>
<tr>
<td>Increase return on investment (GM)</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Increase sales volume</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Increase profit per vehicle</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Increase customer appeal</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Expand markets</td>
<td>0.3</td>
<td>0.0</td>
</tr>
<tr>
<td>Increase sales price</td>
<td>0.0</td>
<td>0.5</td>
</tr>
<tr>
<td>Increase foreign earnings</td>
<td>0.0</td>
<td>0.9</td>
</tr>
<tr>
<td>Lower production costs</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Increase high margin sales</td>
<td>0.0</td>
<td>0.6</td>
</tr>
<tr>
<td>Reduce operating costs</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Lower environmental impact</td>
<td>0.9</td>
<td>0.0</td>
</tr>
<tr>
<td>Lower purchase costs</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Keep labour costs low</td>
<td>0.0</td>
<td>0.9</td>
</tr>
<tr>
<td>Improve economies of production</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Lower gas price</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Improve mileage</td>
<td>0.8</td>
<td>0.0</td>
</tr>
<tr>
<td>Offer rebates</td>
<td>0.3</td>
<td>0.0</td>
</tr>
<tr>
<td>Lower loan interest rates</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Lower sales price</td>
<td>0.8</td>
<td>0.0</td>
</tr>
<tr>
<td>Reduce raw materials costs</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Outsource units of production</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Gas price rises</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Lower Japanese interest rates</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>US gas price rises</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Japanese gas price rises</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Yen rises</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Japanese rates rise</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Backward Propagation

- Formalization of the problem:
  - Linear cost function: \( \min Ax \)
  - a set of non-linear equality and inequality constraints
- ... and we pass the system to Lingo 8.0.
- www.lindo.com
GR Tool
Reasoning about goal models

- Supporting reasoning
  - Different uses
  - Different models
  - Different outputs

G-Reasoning

- Early Requirements
- Late Requirements
- Design
- Design pattern selection
Reasoning about goal models cont.

- Early requirements analysis
  - Allow to analyze the organizational setting intra e inter actor analysis
  - Verify satisfaction of goals
    - A means to discuss with the stakeholders about their goals
  - Find possible conflicts

- Late requirements analysis
  - Evaluate possible alternative functional requirements wrt non-functional requirements (softgoals)
  - Find possible conflicts among requirements
  - Reason about requirements impact over stakeholders goals/softgoals
Actor diagram: Early requirements analysis
Actor diagram: *Late requirements analysis*
Reasoning about goal models cont.

- **Architectural design**
  - Allow to decide among different architectures
  - Find and solve possible conflicting situation among subcomponents
    - Important when sub-component(actors) use the same resources
  - Evaluate sub-part of the design (step-by-step evaluation)

- **Pattern selection**
  - Patterns can be evaluated and selected with respect to
    - Their impact on goals/softgoals
    - Their impact on other patterns
Completness
Reliability
Coordinativity
Redundancy
Other Quality Attributes

Claim
["External Agents can spoof the system"]

Distributivity
Participability
Commonality

Joint Venture
Structure in 5

Other Styles

© P. Giorgini
Reference


- http://www.troposproject.org