AGORA: Attributed Goal-Oriented Requirements Analysis Method

Sep. 11, 2002

Haruhiko Kaiya (Shinshu Univ.)

Hisayuki Horai (Tokyo Inst Tech., CLS)

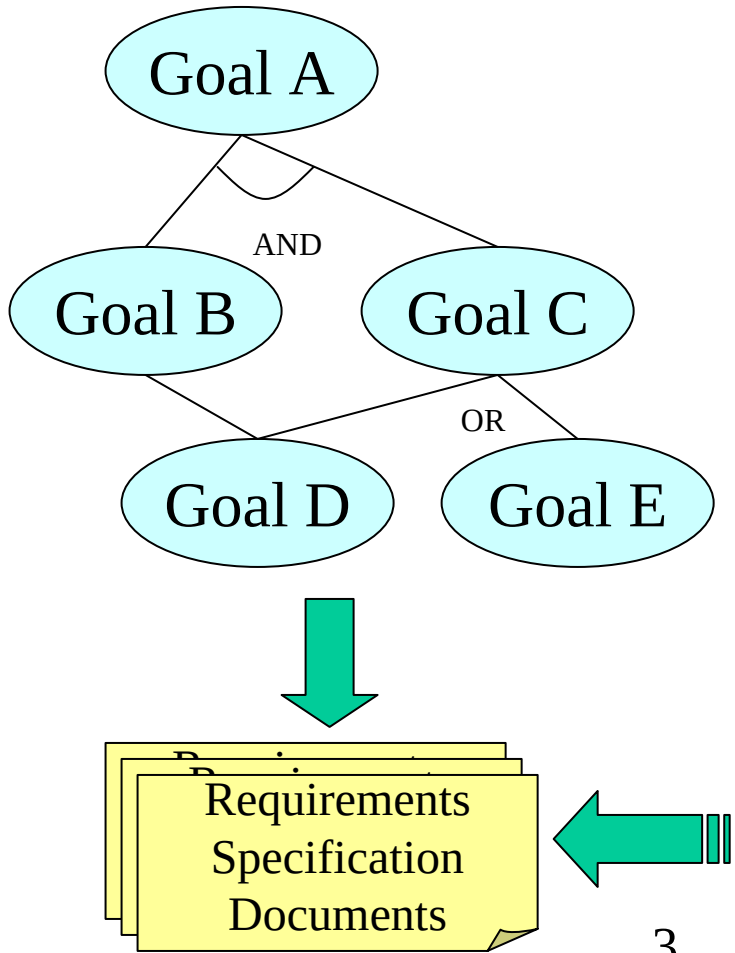
Motoshi Saeki (Tokyo Inst. Tech.)

Contents

- Introduction
- How to analyze requirements with AGORA
  - using Small Example
- How to Predict the quality of a requirements specification document.
  - IEEE Standards 830
  - Quality Model for source code by McCall
- Conclusion and Future Works

GORA (Goal Oriented Requirements Analysis)

- Structural Decomposition of Goals,
  - decrease ambiguity.
  - concrete goals.
- Goal graph: a representation of such structure.
- GORA is useful to elicit, analyze and specify requirements.



Problems of GORA in General

- No clear criteria for goal decomposition.
- No clear way to resolve conflicts among goals.
- No clear way to select a most suitable goal from alternatives.
- No clear way to predict the quality of a req. document during analysis.

AGORA: An extended version of GORA

- Attributed Goal-Oriented Requirements Analysis = AGORA
- Attaching values to nodes and edges of the Goal Graphs
  - Edge: Contribution Value: How the sub-goal(s) contribute to the super-goal.
  - Node: Preference Value (matrix): How each stakeholder prefer the node(=goal)
    - predicting preferences of others.
  - Description of the reason for values (Rationale)

Analyzing Requirements with AGORA

- A simple example about constructing AGORA graph
  - comprehensive explanation of the notation and the procedure can be found in Proceedings.

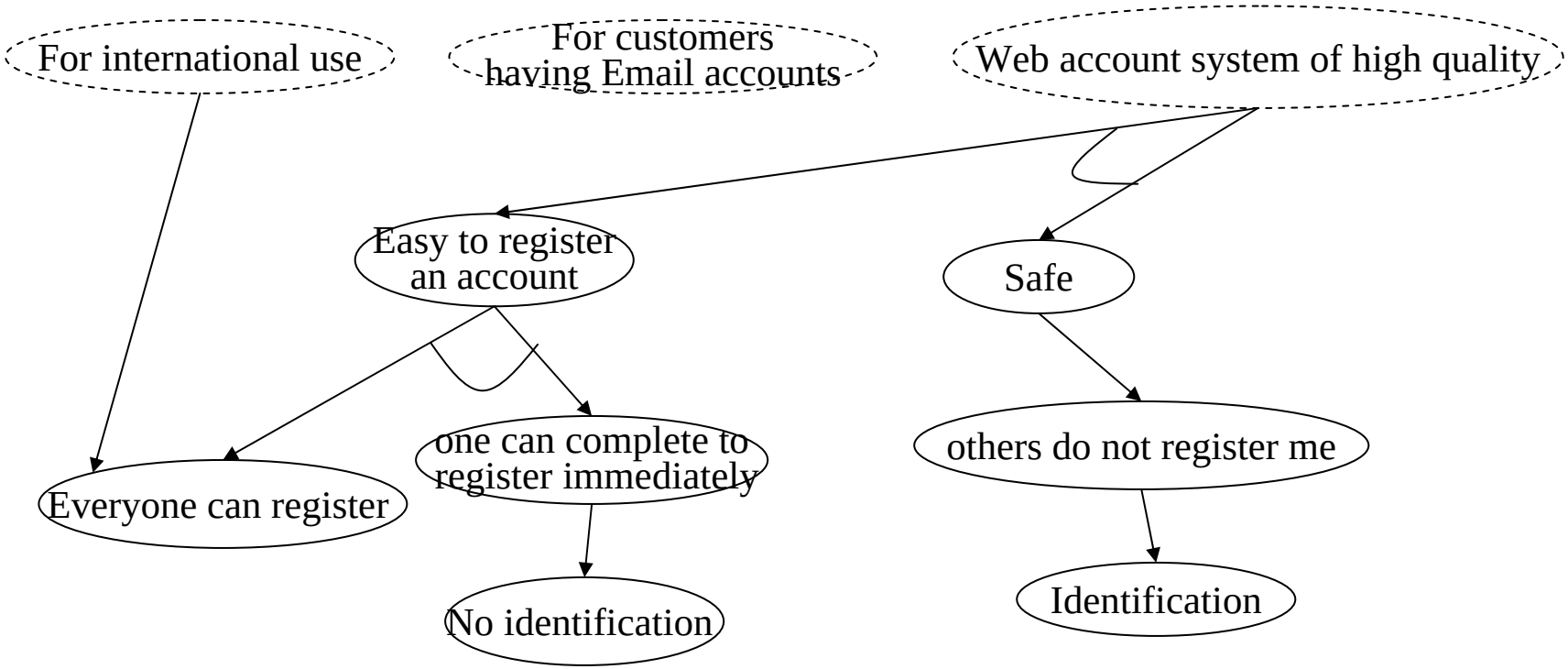
A simple example: WEB account system

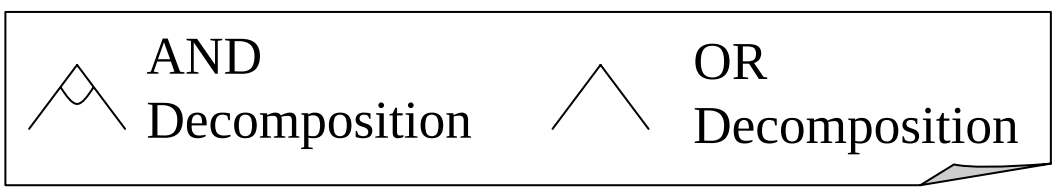
A WEB account sub-system to register customers for e-business or e-learning

- For International use.
- For customers having Email account.
- Of course, High-quality.

They are initial needs, so they become initial goal.

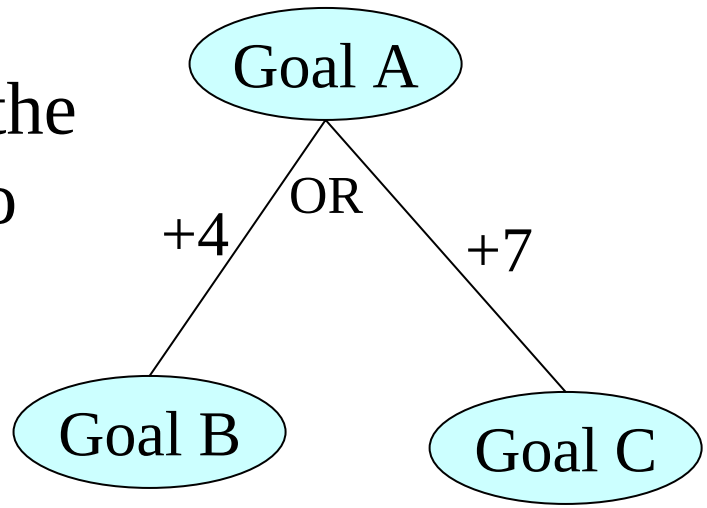
Goal Decomposition (normal GORA)



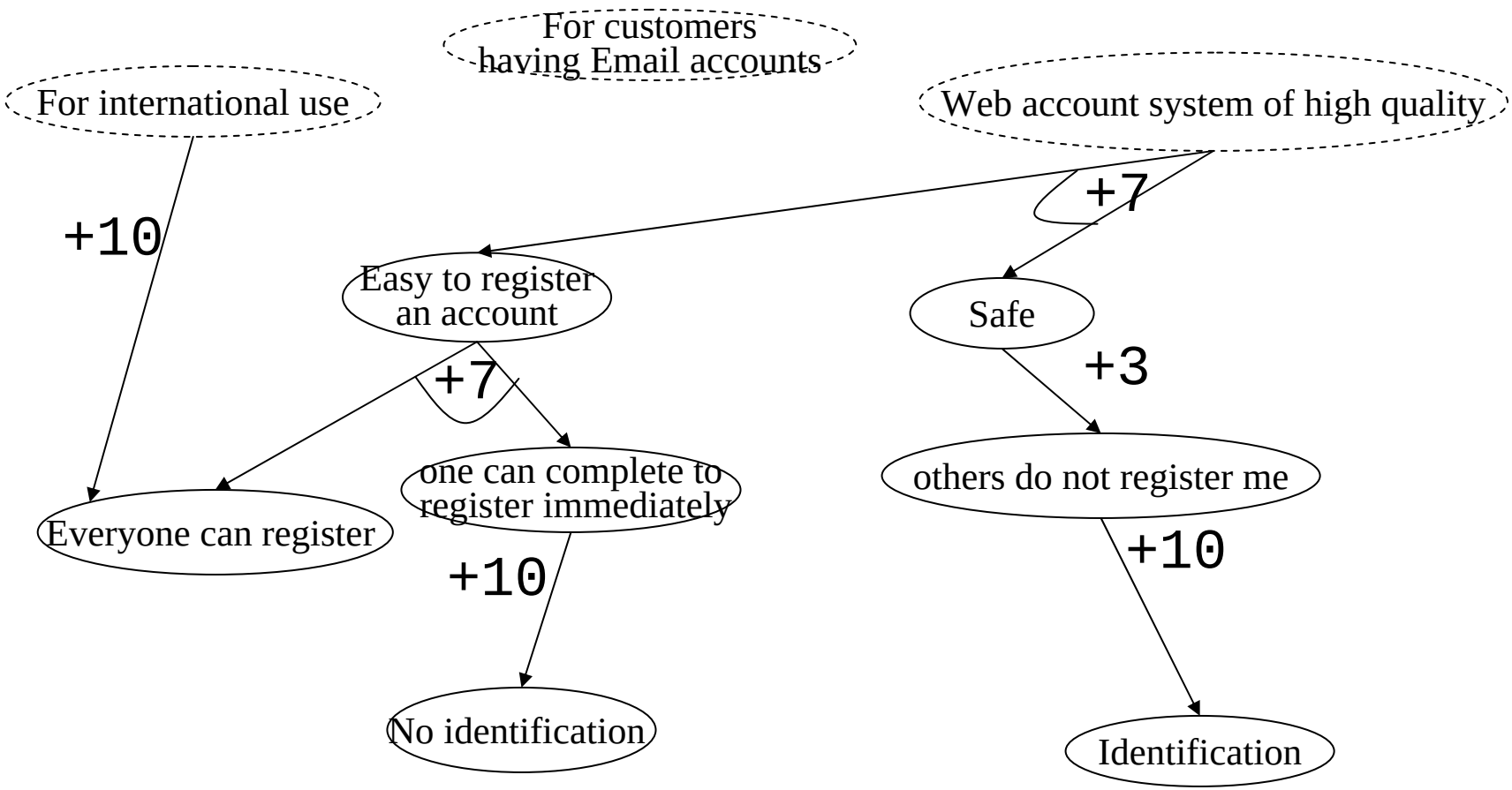


Contribution Value in AGORA

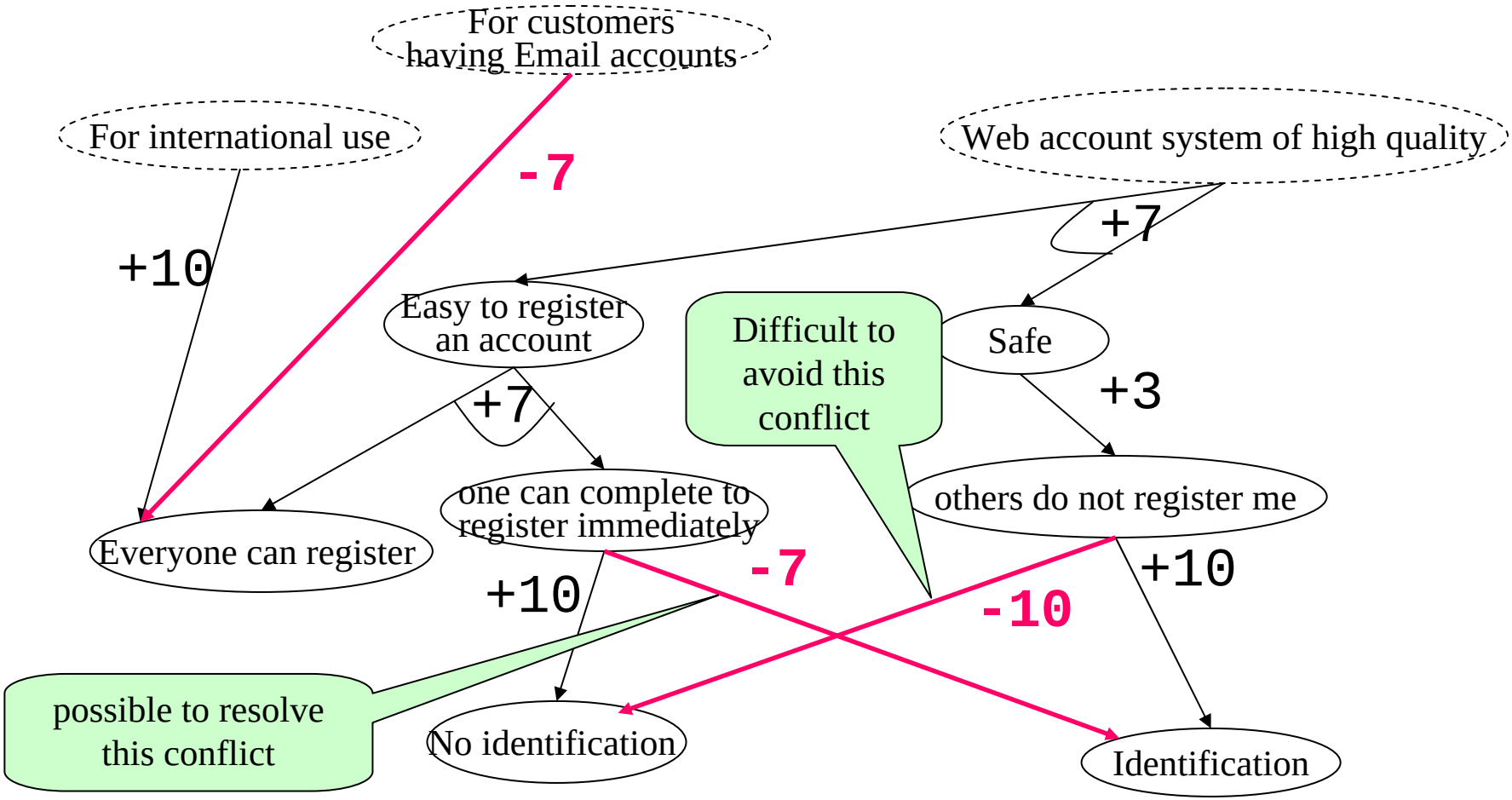
- It is attached to an edge.
- It expresses the degree of the contribution of the goals to the achievement of its connected parent goal.
- Range: -10 .. 0 .. +10 (harmful .. unrelated .. good)



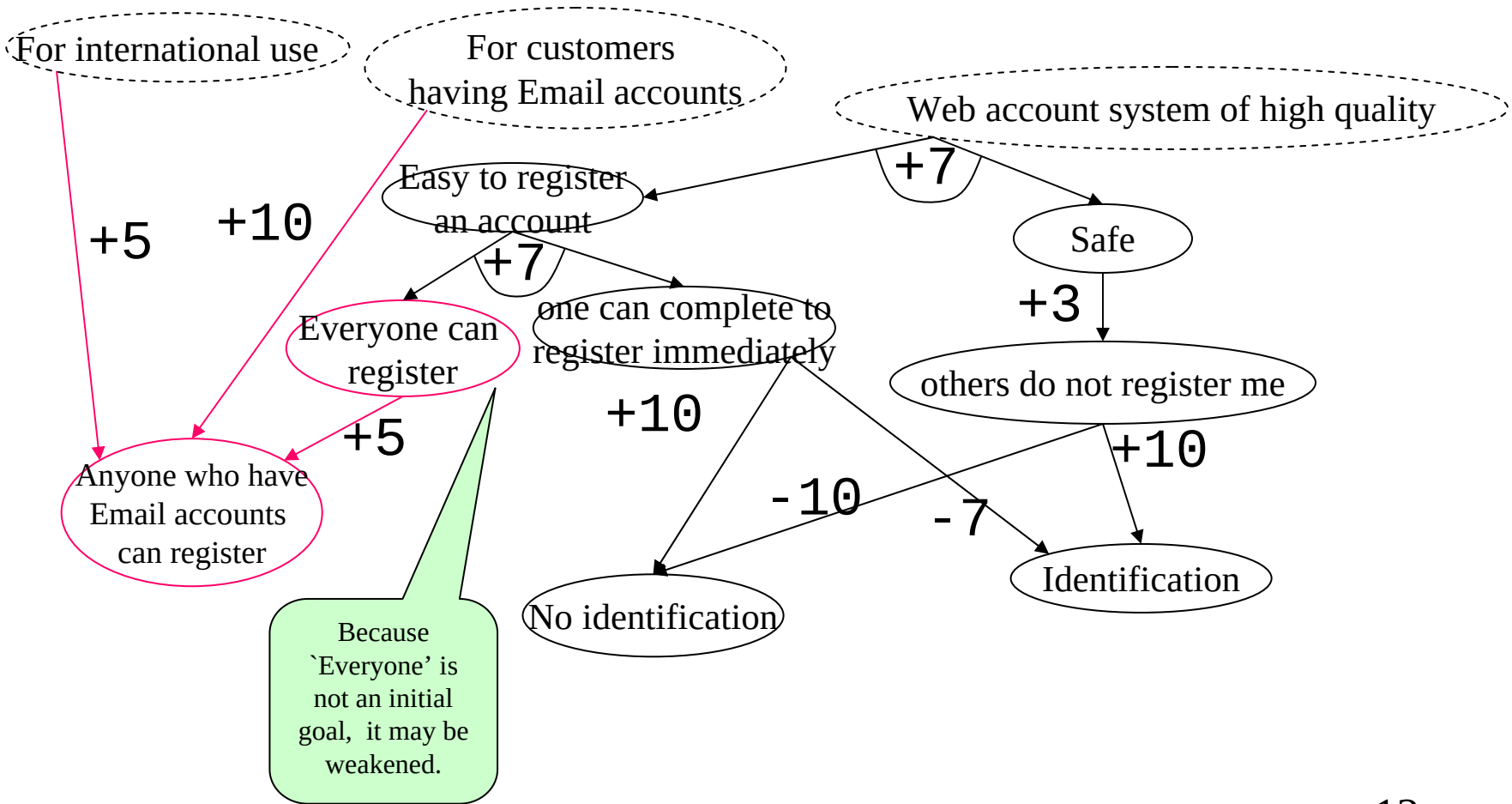
Attaching Contribution Values



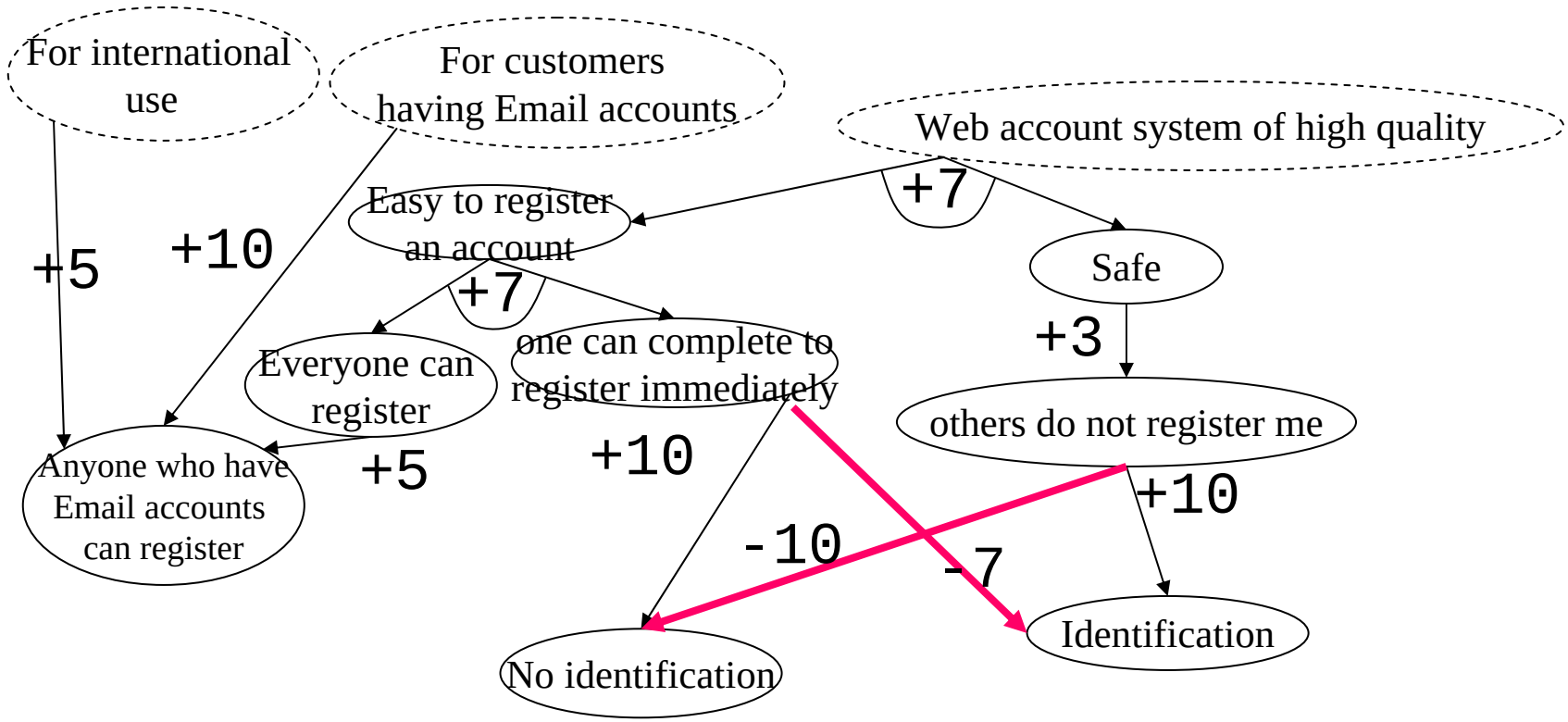
Analyzing Conflicts among goals



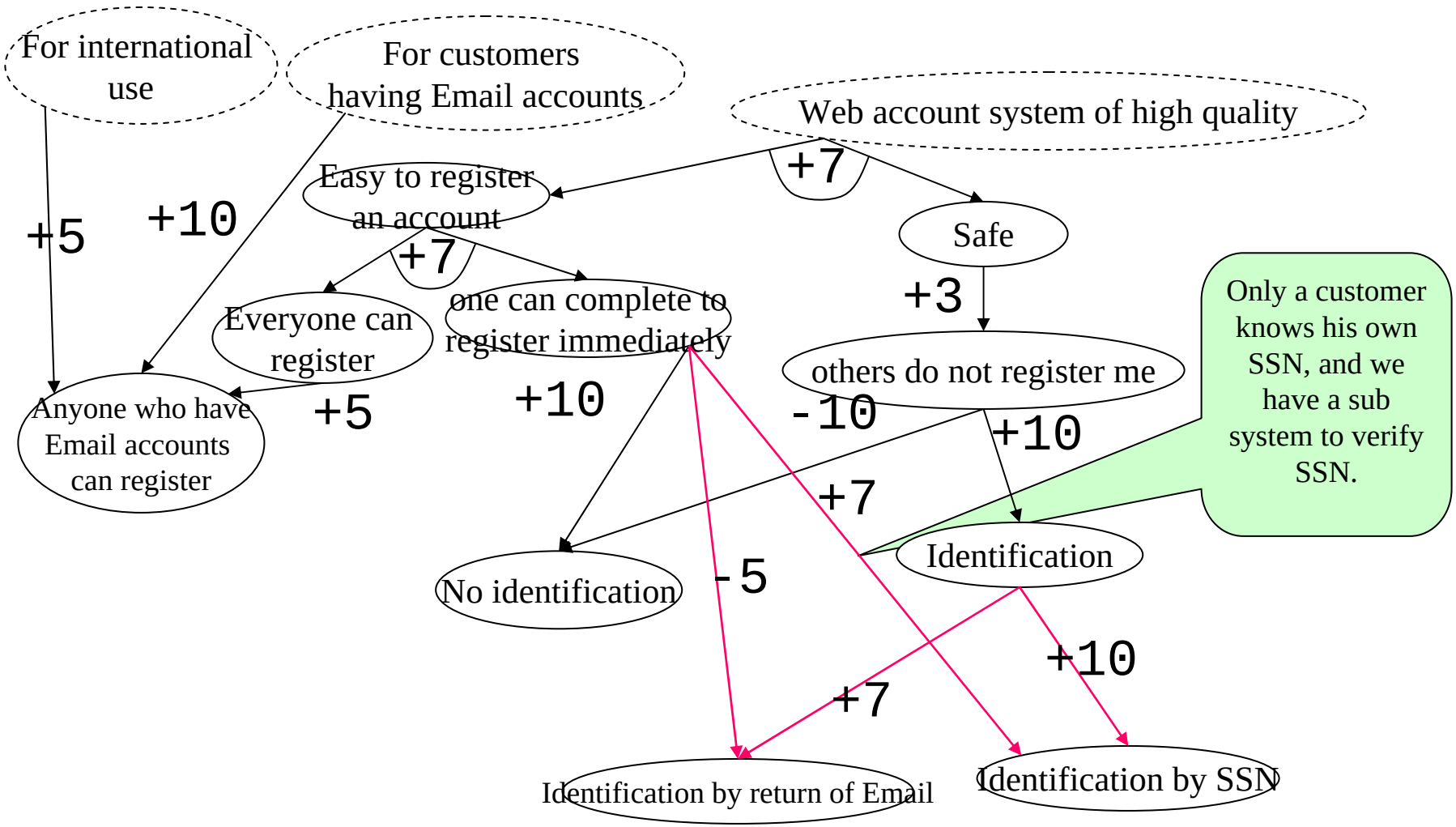
Resolve Conflict by decomposition



Resolve other Conflicts 1/2

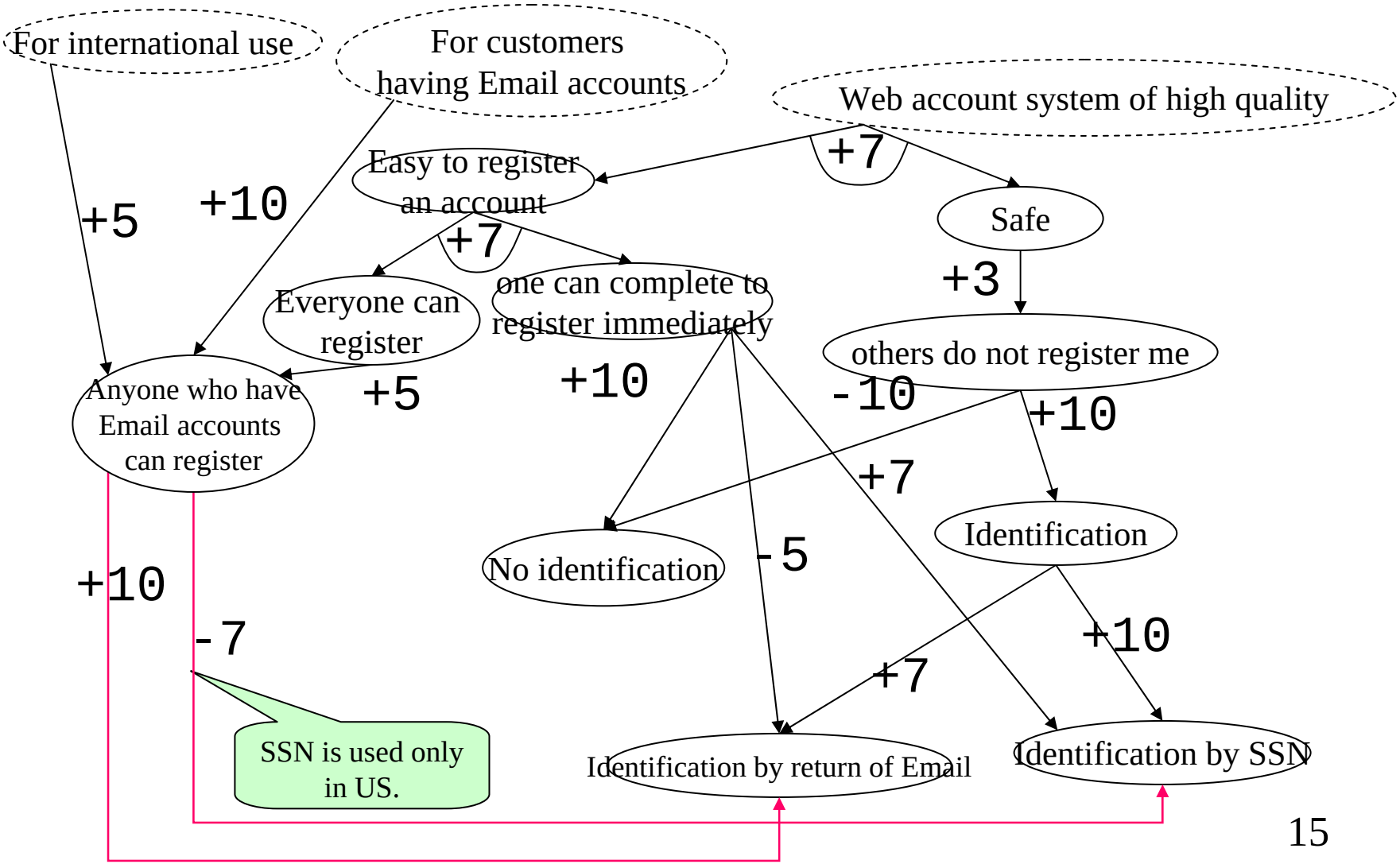


Resolve Conflicts 2/2



SSN = Social Security Number

Analyzing More Conflict



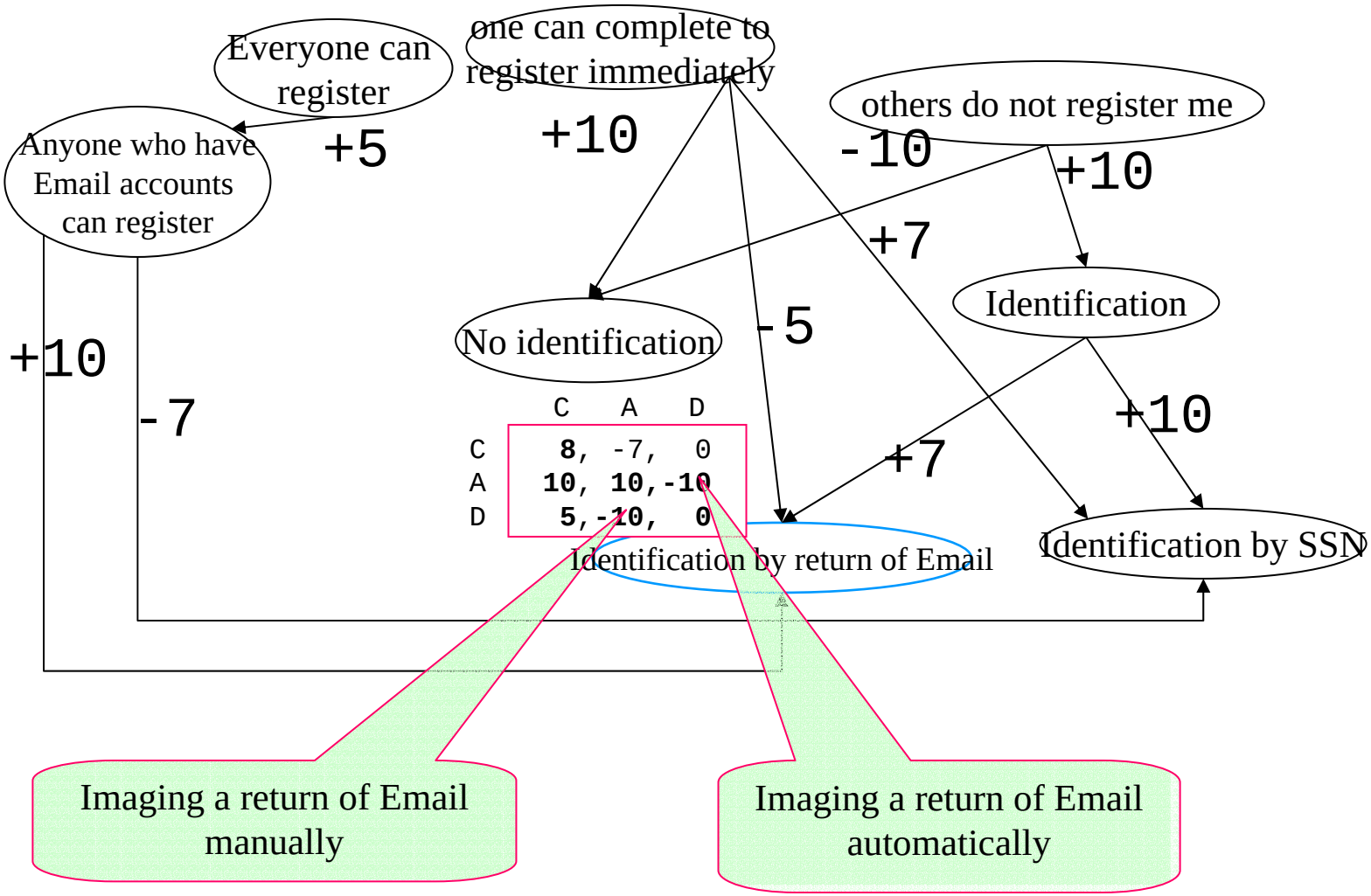
Preference Matrix

- It is attached to a goal.
- It stands for the degree of preference of each stakeholder to the goal.
- Each stakeholder should predict preferences of others.
  - Diagonal values are for themselves.
- Range: -10 .. 0 .. 10 (dislike .. unconcern .. prefer)
- It is not attached to all goals.

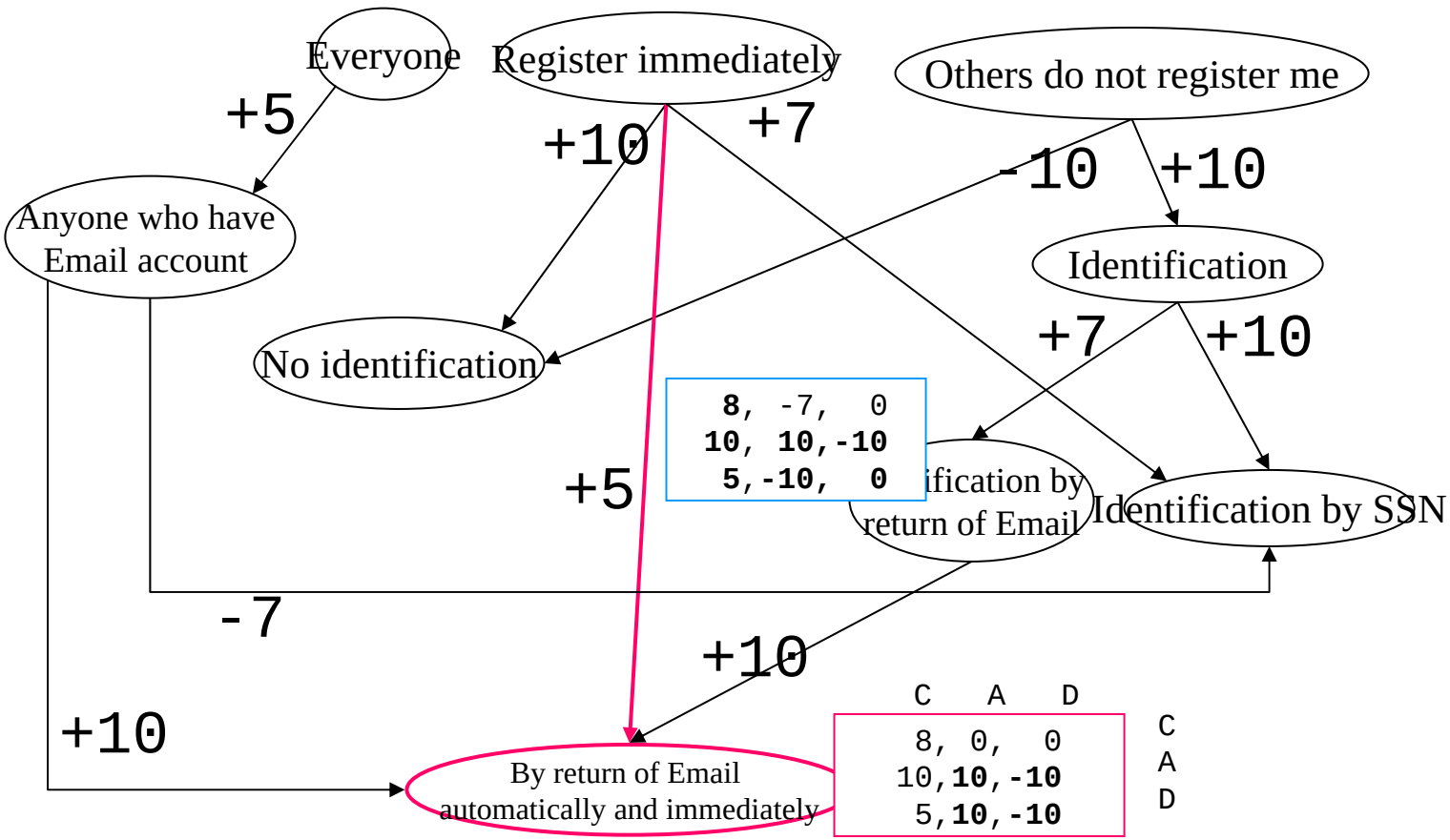
Evaluator \ Evaluatee	C,	A,	D
C,	8,	-7,	0
A,	10,	10,	-10
D	5,	-10,	0

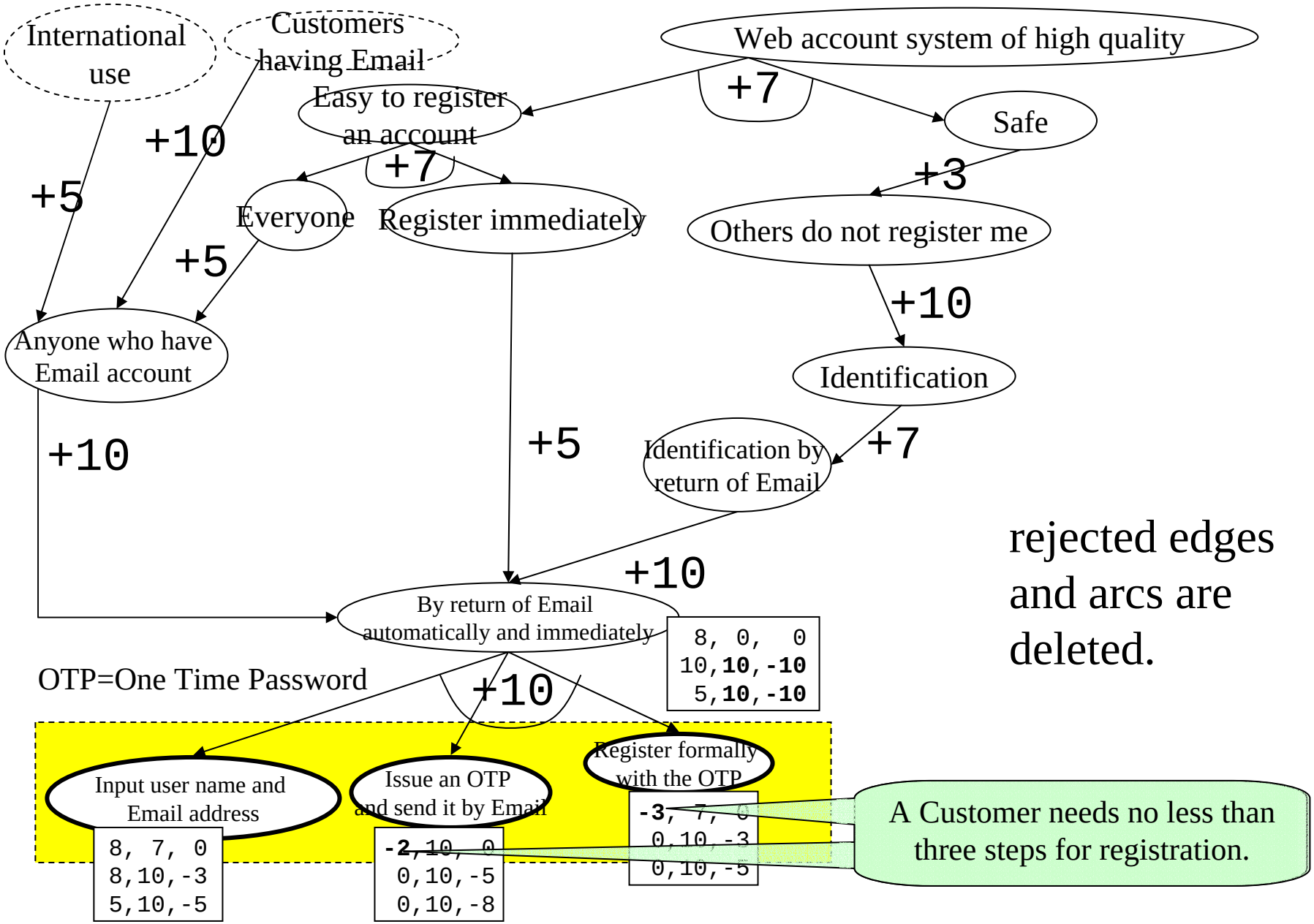
C = Customer
A = Administrator
D = Developer

Example of preference matrix



Resolving misunderstanding by decomposition.





rejected edges and arcs are deleted.

Predicting the quality of Req. documents

- We can not know the quality until the analysis is finished.
- During the analysis, we want to improve the quality of analysis process based on the predicted quality of req. documents.
- In AGORA, we predict the quality of req. documents based on the values attached to an AGORA graph.

Quality characteristics of Requirements Documents

- Mentioned in IEEE 830 standards(1998) and A. Davis's Book.
  - Correctness
  - Unambiguity
  - Completeness
  - Consistency
  - Verifiability
  - Modifiability
  - Traceability
  - Ranked for Importance and Stability
- It is difficult (or impossible) to measure them directly from a document.

Quality Factor by McCall

- McCall categorized the factors of source code quality.
  - Example: Correctness, Reliability, Efficiency.....
- It is hard to measure them directly, so he gave the following equation by calculating the factors indirectly.

$$\text{Factor} = \Sigma\ (\text{Coefficient}_i \times \text{Metrics}_i)$$

- In AGORA, we export this idea into requirements documents.

Quality Factors and Metrics in Requirements Documents

- Quality Factors: using the factors appeared in IEEE standards and Davis book.
- Metrics: defining by the shape of the goal graph and the values attached to goals and edges.

Coefficients for Req. Doc.

Metrics

Factors

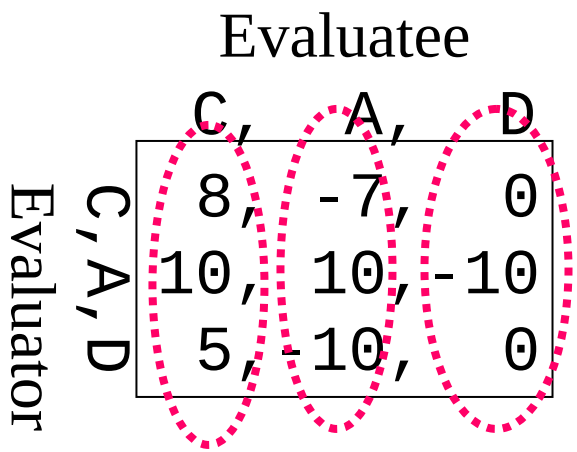
	Sat	Pos	Cup	Vdv	Cov	Hdv	Tre	Con	Rat
Correctness	0.5	0.3	0.2						
Unambiguity				1					
Completeness					1				
Inconsistency		0.6			0.4				
Modifiability							1		
Traceability								0.7	0.3

Example:
Correctness = 0.5 * Sat + 0.3 * Pos + 0.2 * Cup

Set of these values above is an example, the values will be different in each analyst.

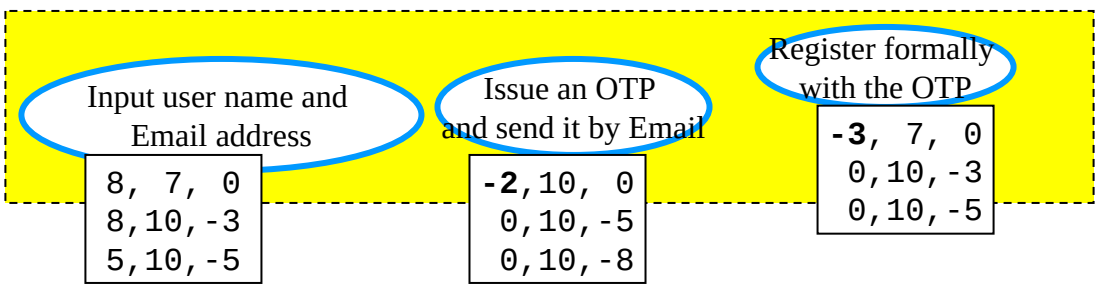
Metrics

- Vdv: The average of variance of vertical values in the preference matrix.
- How stakeholders share same preference or not.
  - **Vdv = 1 – (the average of variance)**



Factor: Unambiguity

- **Def:** A req. document has only one interpretation.
- This factor is directly calculated from the value of Vdv.
- Vdv= 0.14 in the following leaves, they are very ambiguous.



Summary

- Propose a notation and techniques for Extended GORA, AGORA.
- Propose a way to predict the quality of req. document using AGORA graph.

Future works

- Propose a seamless way to convert goal graphs to a requirements document.
  - One of the idea is to regard leaves in a AGORA graph as a Use cases in a Use case diagram.
- Method and Tool to support AGORA.
  - Communication between stakeholders and analysts.
  - Patterns and Heuristics for constructing a goal graph.
- Method to decide the values in a AGORA Graph.

That’s All, Thank you