

Goals, Policies and Requirements Completeness

White Paper for NSF/NIST Workshop on Policy Specification

Colin Potts

College of Computing, Georgia Tech.

December, 1994

1. Requirements Completeness and System Flexibility

Policies are rules or principles according to which enterprises operate. They constrain the freedom with which individuals or organizations may act when performing their responsibilities, and they constrain possible implementations of information systems. Examples of policies include rules governing who may view certain information (security and privacy policies) and rules for avoiding potential hazards (safety policies).

It is important to treat the definition and operationalization of policies as an integral part of the design process. At Georgia Tech, we have been exploring design methods for ISs that incorporate novel approaches to requirements analysis. Our position is that the requirements for a new system should be regarded as the fulfilment or operationalization of prior enterprise goals. The concept that physical or information processing operations are operationalizations of underlying goals or intentions is familiar in the domain of human-computer interaction (Norman, 1988), where a conceptual gap often exists between what a user wants to do and how the user must interact with the system at the level of physical operations and perceptual events. A similar translation process must also occur at an enterprise level between what the customer organization is trying to achieve and the functions and behaviors provided by a computer-based system (Figure 1).

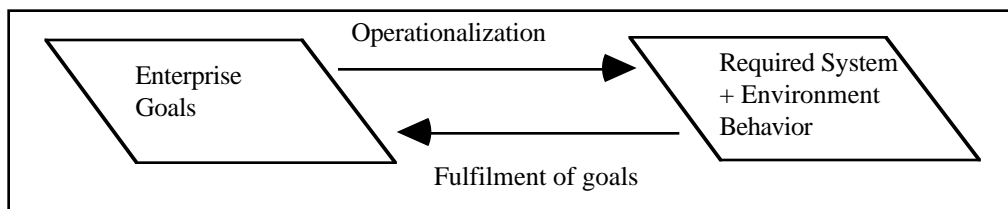


Figure 1: The 'requirements gap' between enterprise goals and the requirements for system/environment behavior. (Adapted from Norman's [47] model of human-computer interaction.)

Thus the starting point for the analysis is a set of goals, which are then refined into operational behaviors of a required system and its environment. The refinement process is sketched in Figure 2, which shows a sequence of expressions and refinement steps. Each step produces or revises a goal specification, design specification or set of scenarios. The approach elucidates the observance of policies by the specified system, especially through the systematic consideration of obstacles that can thwart the fulfilment of goals (e.g. policy violations or unnecessary inflexibility of the system in its organizational environment).

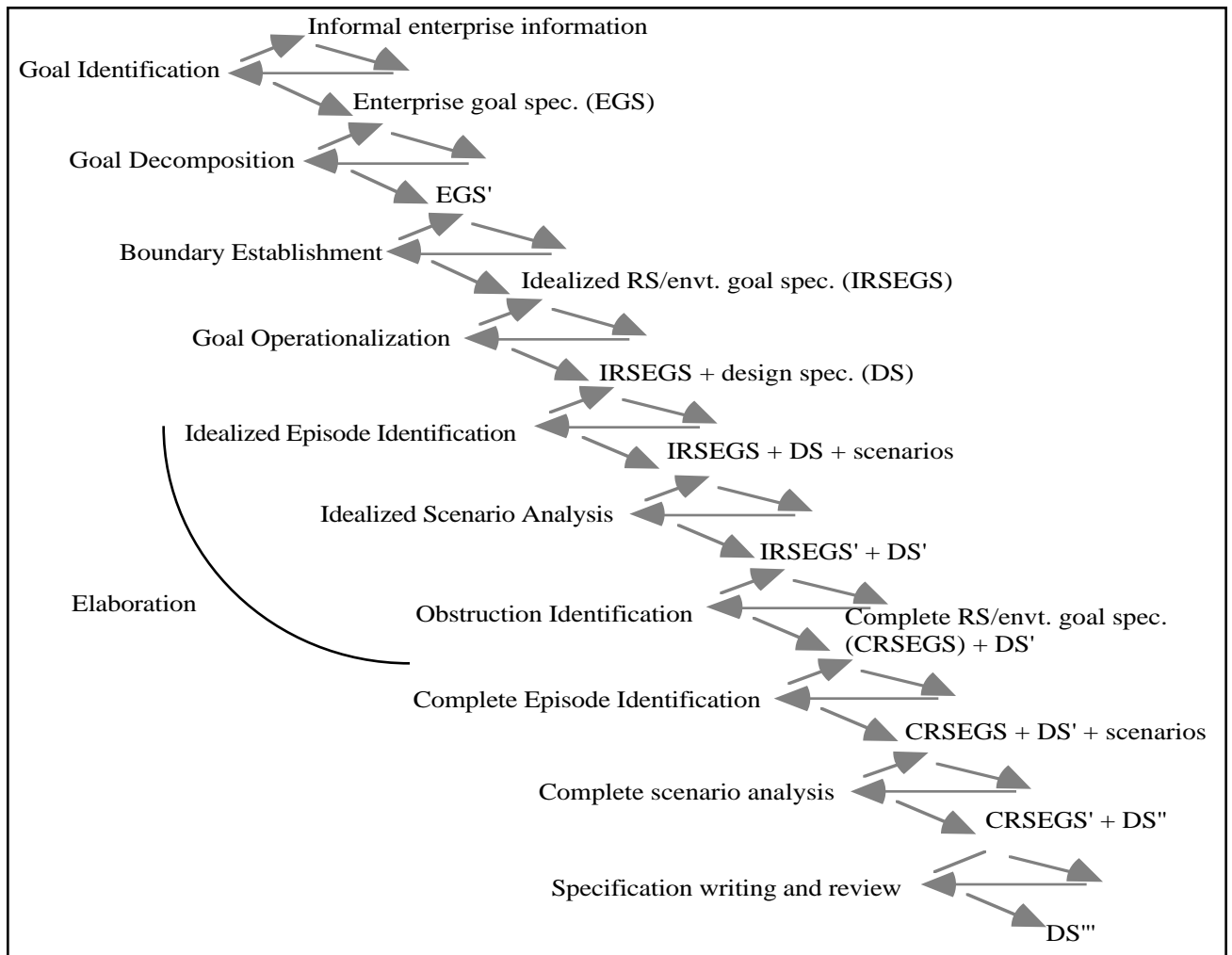


Figure 2: Strategy for performing goal refinement and scenario analysis. Each stage represents an application of an inquiry cycle consisting of expression (specification of goals, operational behavior or scenarios), discussion (challenging or reviewing expressed information) and refinement (decomposition, elaboration, boundary establishment or operationalization).

Our approach to goal elaboration currently starts with a discussion in which the following recurrent questions about obstacles can be asked about all possible goals: (a) *Failure occurrence*. Can this goal be obstructed, and if so, when? (b) *Failure consequence*. If this goal can be obstructed, what are the consequences?

The heuristics for identifying failure occurrence include identifying the feasibility of exception handling in the following obstruction situations: (a) *Prerequisite failure*. Can another goal be obstructed that is a prerequisite to this goal? (b) *Replicate collision*. If the goal deals with one object among a population of similar objects, can these replicate objects contend for limited enterprise resources, which, if exhausted, would obstruct the goal? (c) *Replicate confusion*. If the goal deals with one object among a population of similar objects, can these replicate objects be confused, thereby obstructing the goal? (d) *Actor failure*. Can the actor responsible for achievement of this goal fail or become otherwise unavailable, thereby obstructing the goal?

Deciding *which* obstacles to deal with is largely informal, depending upon reasoned discussion, but in the case of some questions it could involve statistical analysis, safety techniques such as fault-tree analysis and planning techniques.

Having identified obstacles for further analysis and possible goal elaboration, there are several principles that may be used for identifying which operational scenarios should be explored. (a) *Normal cases and goal thwarting*. Given the goal dependencies a minimal set of operational scenarios should be constructed that covers all the goals (see the precedence honoring rules discussed under the next research issue). (b) *Single Obstacles*. Operational

scenarios with single goal obstacles for the meeting scheduler include unscheduled meetings because of unfeasible schedules or room unavailability, failure of a potential participant to respond to the Initiator with meeting time preferences (in the case of the Email design scenario), failure of a participant to update his or her calendar (Calendar design scenario), etc. (c) *Combinations of obstacles*. An example heuristic might be to combine a standard obstacle with a contention or object confusion obstacle. Operational scenarios help raise questions that drive completeness-increasing refinements. Some possible elaborations stemming from a discussion about the meeting scheduler are contained in Table 1.

Obstacle / combination	Defensive actions	Mitigation actions
Unfeasible schedule	Organizational (build more rooms; abandon meetings, etc.)	(1) Prioritize normal participants and schedule high-priority participants first. (2) Schedule as many normal participants as possible, (3) Initiator asks for additional preferred times from participants
Participant does not respond (Messaging)	(1) Organizational (Culture of replying to requests). (2) Remind periodically before scheduling date passes.	(1) Make decision anyway ignoring non-respondent. (2) Give non-respondents a choice of valid time slots.
Participant does not keep calendar up to date (Calendar)	Organizational	Confirm meetings immediately schedule is calculated.
Facilities change + confusion over identity of the room that was chosen.	n.a. (combination)	Confirm every room booking with Facilities organization whenever a room is closed.

Table 1: Example obstacles to goal achievement and possible defensive and mitigation goals.

We define *episodic goals* as those goals that are useful in identifying episodes, the building blocks of operational scenarios. Our experience suggests that episodic goals are to be found near the middle layers of the goal hierarchy, but we need to develop sharper heuristics for identifying the appropriate levels for practical problems. It is possible that the “correct” level may depend on the type of questions being investigated by the operational scenarios. Episodes usually have variants that represent alternative ways in which the episodic goal is achieved or thwarted.

4. Evaluation

Most of the tentative suggestions for refinement heuristics originate from two current research projects that are examining the specification development of two commercial systems: one a communications infrastructure product, and the other a commercial Internet service (Potts et al., 1995).

We have stressed the analytic process, not the role of collaboration in specification development. Nevertheless, we acknowledge the importance of collaboration (and other contextual/social factors such as negotiation of goals among holders of opposing viewpoints), which we are investigating in ongoing empirical projects at Motorola and NTT (Potts et al., 1995). These projects are investigating the collaborative processes that affect specification development and the effects of tool deployment on requirements convergence, team working memory, and the ownership of requirements-related information. The NTT project has made explicit use of a watered-down goal refinement method and inquiry-based requirements reviews.

References

Potts, C., K. Takahashi, J. Smith and K. Ota, “An Evaluation of Inquiry-Based Requirements Analysis for an Internet Service, to appear in Proc RE’95, York, 1995.