Modelling the Interplay of Conflicting Goals with Use and Misuse Cases

Ian Alexander

Independent Consultant

Ian.Alexander@ScenarioPlus.org.uk

Summary

Business goals often conflict. Conflicting goals can come from within an organisation, or may appear as external threats: goals may be friendly or hostile. Relationships between goals therefore need to go beyond conventional and/or inclusion. This paper suggests an economical set of four relationships: threatens, mitigates, aggravates, conflicts with. It gives examples of three general types of situation in which these relationships between friendly and hostile goals help to define business situations. The approach has been applied in a trade-off study.

Introduction

Businesses, like all human activities, are driven by people's goals and ambitions. Differing goals often bring people into disagreement. Even goals held by the same individual frequently turn out to conflict with each other, directly or indirectly, when their consequences are evaluated. And even when goals are shared within an organisation, there are often people outside it who actively oppose those goals for commercial, political, or personal reasons.

Both goal modelling [e.g. van Lamsweerde 1998] and use case modelling [Jacobson 1992] are well-tried approaches in the software domain, and both have been applied to some extent to business processes. Jacobson considered the roles of different 'actors' in a process; van Lamsweerde also considered the place of 'obstacles' to desired goals. However, neither have given much attention to intentional opposition.

A **goal model** is an organized structure of (usually functional) goals, typically forming a tree or network, with AND and OR links between goals and subgoals. Goal models do not necessarily document timing constraints, context, or the roles involved.

A **use case** is an organized branching structure of steps (activities taken by named roles) in time-sequence, together with supporting information such as preconditions, defining both normal and exceptional sequences of events, to achieve a named goal [Cockburn 2001]. A **use case model** is a network of use cases, related by inclusion and possibly other links. Thus a use case model forms a goal hierarchy together with enough related information to be useful as a specification.

However, in business there are often conflicting goals, for many reasons. For example:

- Legitimate stakeholders have different viewpoints, which may conflict.
- The business may be threatened by external agencies such as rival companies or activists determined to damage the business.
- Different parts of the business may rightly desire to improve their own performance, but local optimisations may cause undesirable side-effects in other parts of the business.

These may prove damaging if not identified and presented for resolution: the first step in conflict negotiation is always to know what the conflict is about. An adequate goal model of a business process must therefore be capable of describing potential conflicts.

This paper illustrates a simple way of describing goal conflicts. It represents legitimate goals as use cases; hostile goals as misuse cases; and interactions between goals as relationships of the two standard types (includes, has exception), and just four conflict analysis types (threatens, mitigates, aggravates, conflicts with).

Since the purpose of modelling a business process is to create shared understanding, it seems important that the concepts used should be simple and readily understood by people who are not analysts. The concepts (goal, hostile goal, threat, mitigation, aggravation, and conflict) are arguably familiar to business people, and can be explained in a few minutes.

Related Work

Several approaches to goal modelling exist. The KAOS method models goals with a precise version of classical AND-OR trees [van Lamsweerde 1998]. Use Cases organize functional requirements as steps within scenarios, each with a functional goal [Jacobson 1992]. Use Case models can therefore form goal hierarchies when use cases include other use cases, and when their structure indicates how those inclusions are to be combined – essentially, either ANDed in a time-sequence, or ORed as alternative time-sequence paths. Numerous task-modelling approaches, derived from User Interface work, also exist [e.g. Alexander 1998]. However these approaches do not directly support the modelling of conflicting requirements, nor the causes of conflict such as hostile agents.

Goal-Obstacle analysis goes further by considering obstacles in the way of desired goals and the possibility of thinking out exception goals as a way of 'surfacing' new requirements [van Lamsweerde 1998, Anton & Potts 1998], but these obstacles are treated as passive, whereas hostile agents can all too actively oppose and counter goals.

The i* approach to non-functional requirements [Chung 2000] introduces the idea of a 'soft goal' that complements the idea of a desired functional goal. A soft goal is something non-functional, like the desire to be safe or secure or to arrive quickly, that can be met by a product of sufficient quality. Goals, soft goals, and related items form a dependency model, linked by depends-on and other relationships. However this approach does not explain the reasons why soft goals may be desirable: why safety or security may be threatened, for instance.

Misuse Cases introduce the needed element of active opposition to a goal [Sindre & Opdahl 2000]; similar concepts have been proposed for eliciting security requirements [McDermott & Fox 1999] and for safety [Allenby & Kelly 2001]. However this approach stops short of exploring and defining the relationships between hostile and desired goals, which are of novel types.

Direct Conflict of SubGoals

The simplest situation is perhaps where a business goal is to be achieved through two subgoals, both considered desirable, which directly conflict and need to be traded-off against each other. For example, subgoal A may involve increasing the number of sales outlets (to raise turnover), while subgoal B involves decreasing it (to maintain an exclusive marketing profile).

We can straightforwardly model this situation by treating the goal as a high-level use case and the subgoals as included use cases, following [Cockburn 2001]. The UML (Unified Modeling Language) allows additional relationships to be defined as stereotypes. A bidirectional relationship, 'conflicts with' makes it easy to describe the business situation (Figure 1).

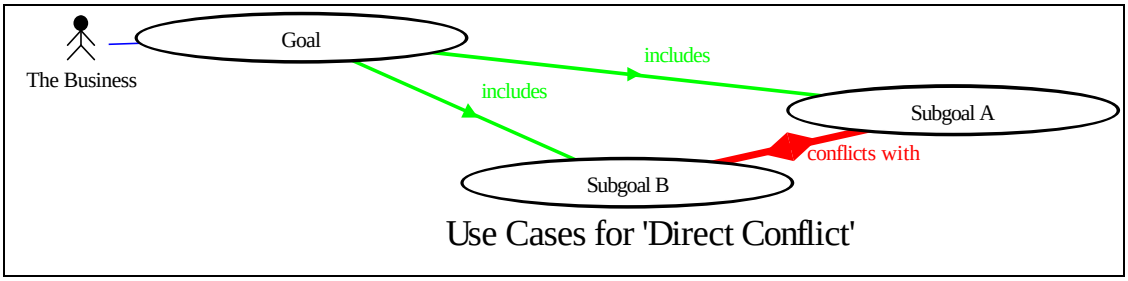

Figure 1: Directly Conflicting Goals

The Threat / Mitigation Cycle

A second common situation is where a business goal is threatened by a hostile goal which is desired by some hostile agent. This is familiar in the real world where businesses compete and are threatened in many ways, most obviously by attacks on the security of electronic commerce websites.

Businesses respond to threats by specifying appropriate responses; achieving such a response is a subgoal (which would not otherwise have been desired). The responding subgoal attempts to mitigate the threat, for example aiming to prevent a hacker from breaking into a website. The hostile agent may respond to the business' response, such as increased security, by developing more sophisticated attacks; each of these is a hostile subgoal. The business may respond to these subgoals in turn by improving its own defences, such as by developing improved security measures.

This situation is a classic maximise/minimise (minimax) approach, similar to that used by chess or go players: white's best move is to work out black's best move, and neutralise it. The threat-mitigation cycle can be represented in goal form quite directly by introducing stereotypes for 'threatens' and 'mitigates' relationships, and by labelling hostile goals as black 'Misuse Cases' [Sindre & Opdahl 2000]. The general pattern is illustrated in Figure 2; there may of course be several subgoals on each side, rather than just one.

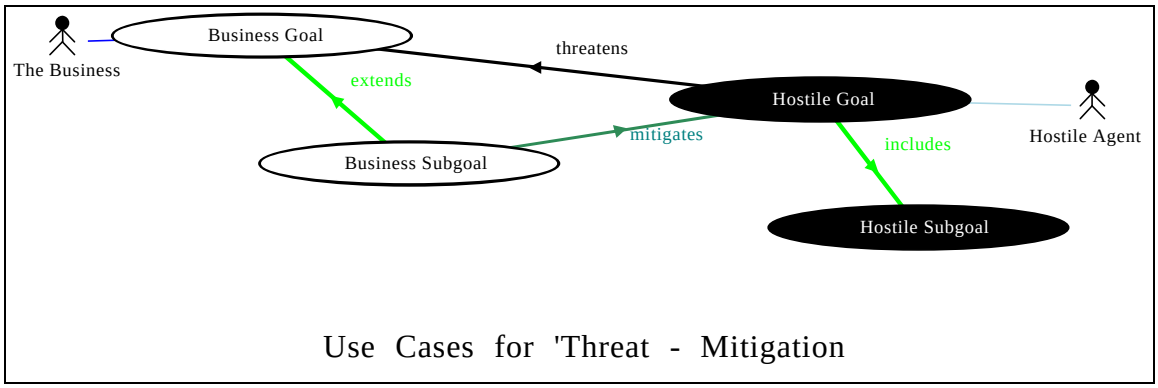

Figure 2: Threat and Mitigation

Indirect Conflict through Mitigation / Aggravation

The third and last type of situation to be considered here is the kind of conflict where two subgoals, both felt to be desirable for some reason, have opposing effects on a hostile goal. One tends to mitigate the hostile goal's undesirable effects; the other aggravates them. This causes an indirect conflict, which may be less easy to detect and trade off than a direct one.

For example, the subgoal 'make the business secure' may mitigate the threat of intrusion, but it aggravates the threat 'make access so difficult that customers go away'. Conversely, the subgoal 'make the business accessible' may aggravate threats to security. There are

many instances in business where desired goals do not necessarily and logically conflict, but given the limitations of current technology they cause trade-offs in practice.

The resulting general pattern is illustrated in Figure 3. Again, in a real situation it is likely that there are multiple conflicts and threats, not just one. Indeed, all the situations described here can occur together.

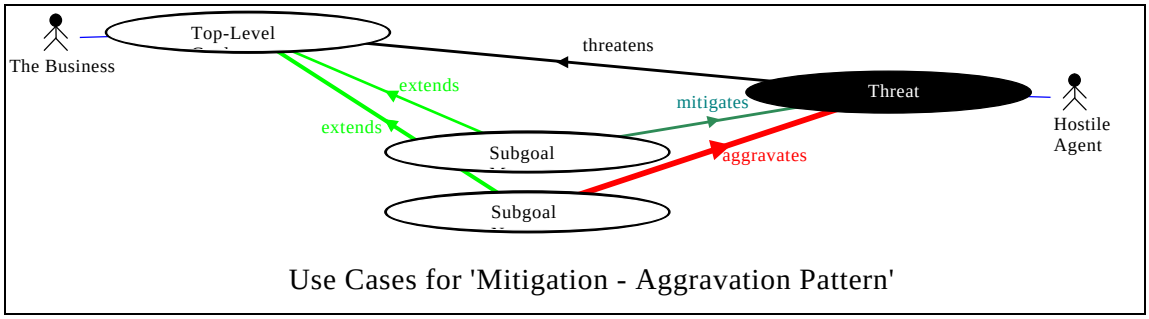

Figure 3: Mitigation and Aggravation

Discussion

A simple goal notation, such as this one based on use and misuse cases, seems capable of expressing a wide range of business relationships and dynamic situations. These may be closer to game-play (and military strategy) than to conventional business process models of the kind that emphasize rigid sequences of actions and decisions. The use case approach (handled more conventionally) is capable of describing sequential behaviour as primary, alternative, and exception scenarios [Alexander 2001] but is not limited to that.

Diagrams like those in this paper make clear in a non-technical way that there is not necessarily a 'right answer', nor a single definite sequence of events that is guaranteed to achieve a high-level business goal. Instead, the diagrams may be useful in helping business people and engineers come to grips with situations where different people justifiably hold differing views; where threats cannot be assumed to be totally neutralized; and where equally desirable subgoals cannot all simultaneously be satisfied (see illustration in Figure 4 below). The diagrams were produced using a specially-developed toolkit that permits editing and filtering so that diagrams can be presented gradually, e.g. without misuse cases, or one or two types of relationship at a time [Scenario Plus 2001].

The question of how many types of relationship to employ is difficult. Ideally there are few to make the 'language' easy to learn and to apply; but conversely there should be many to allow people to express their meaning with precise variations.

One area where more types of relationship may be desirable is in ways of neutralizing threats:

- 'Mitigation' (Latin: Mitis = soft, Agere = to make) means to make (a threatened effect) less severe. This implies it neither prevents a threat, nor renders it less likely.
- 'Prevention' (Latin: Pre = before, Venire = to come) means to make something impossible before it even starts. Some classes of threat can indeed be excluded completely, e.g. a secure site with no remote access connections cannot be hacked into with any amount of knowledge of operating systems or passwords (though its security might be compromised in other ways).
- 'Making less likely' is conceivable but difficult to demonstrate, at least until much design detail is known.

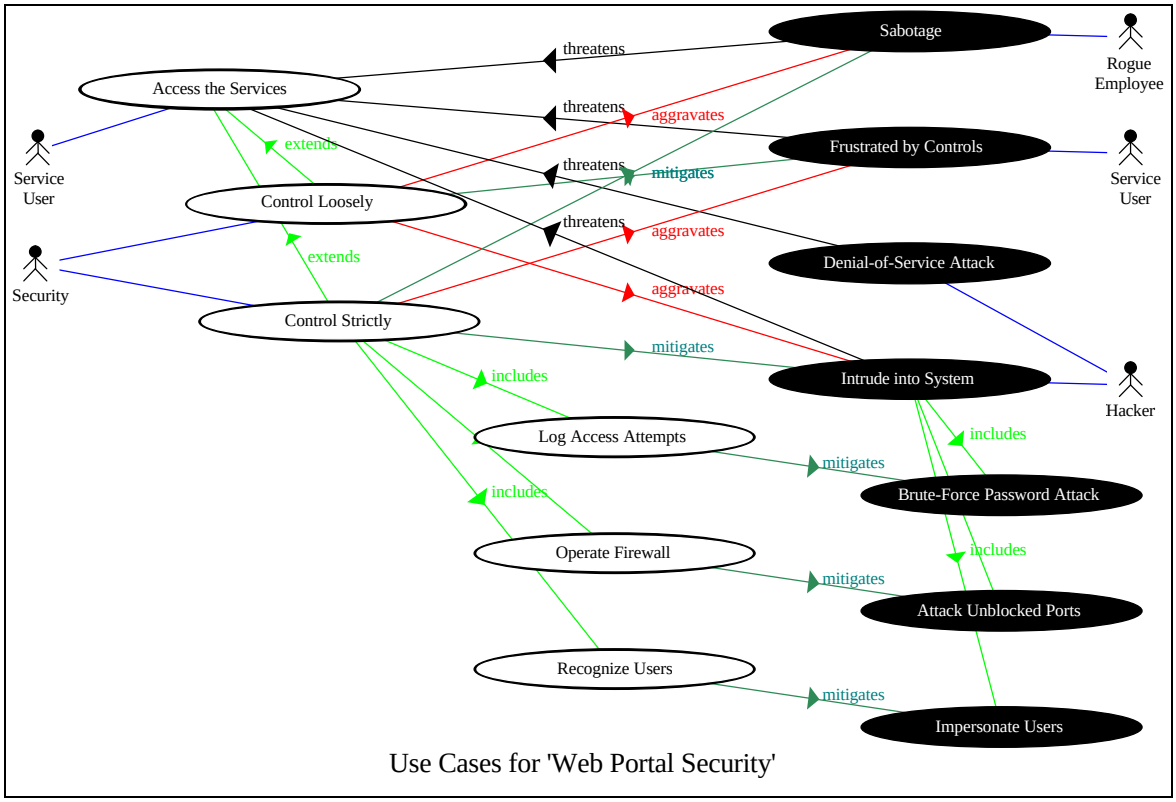

Figure 4: Describing Goal Conflicts with Use & Misuse Cases

The approach has been applied to explain some of the actual and potential conflicts in a railway trade-off workshop [Alexander 2002]. Once it became clear what the interactions were, the designers present in the workshop were able in a few minutes to propose three candidate solutions to a problem that had up till then appeared intractable. None of the stakeholders in the workshop had previously been exposed to goal models or use cases.

Benefits of the Approach

The approach is simple, visual, and compact, using a small set of constructs. It should therefore be easy to learn.

It uses ordinary UML Use Case notation, extended with stereotypes for the conflict-handling relationships. UML is becoming a dominant paradigm; using it enables goal modelling to move into the mainstream of business modelling and system development.

The approach makes explicit the handling of goals and agents, whether desired or hostile.

No special tools are needed. The graphics can be generated by hand, or with ordinary drawing tools such as PowerPoint or Visio. Only modest customisation was needed to use the approach with the requirements management tool DOORS, as demonstrated by the Scenario Plus toolkit. Similar tool support could probably be provided in other modelling environments.

The approach makes no assumptions about domain, so in principle it may be applied to a wide range of business situations. Possible applications include:

- Business process modelling
- Domain modelling
- Requirements elicitation
- Conflict resolution
- Trade-off analysis

Limitations of the Approach

Not everyone likes use cases.

The approach is most suitable for functional goals. Misuse Cases can certainly suggest the need to mitigate threats, such as specific safety hazards, and elicit suitable candidate responses, such as subsystem functions able to mitigate those threats. But the approach does not directly express non-functional or 'soft' goals, such as to be safe, at least if the use cases are to remain classical goals for active scenarios. Alternative approaches such as i* [Chung 2000] model soft goals directly but do not help to define scenarios (sequences of tasks to achieve a goal) and are not compatible with use case or scenario approaches.

Hierarchies of 3 or more levels of goals (up to 5, following Cockburn) can be represented – they are displayed using the convention that high-level goals are on the left. But it is awkward to display a large number of goals with the chosen convention, as there can be only one goal of a given level in a given 'row' (horizontal slice of diagram): two goals at the same level are represented one above the other. In fact in the current implementation of the Scenario Plus tool, there are never more than one positive goal and one negative goal (i.e. one use case and one misuse case) on a given 'row'. This is simple to handle, and reduces the visual clutter of crossing lines, but imposes a practical limit of about 15 goals per diagram (see illustration above) – enough for many purposes. The tool also allows goals to be filtered by level (e.g. you can view only goals at levels 'High' or 'Overview').

Conclusions

It is perhaps too early to draw many conclusions about the value of the use/misuse case approach to goal modelling. The approach has practical value, and may be applicable to problems of many kinds in different domains.

Dedication

This paper is respectfully dedicated to the memory of Dr. Maxim Khomyakov. I am grateful for his kindness in visiting me here in London and his gentle encouragement for my work.

References

[Alexander 1998: *A Co-Operative Task Modelling Approach to Business Process Understanding*, Workshop on Object-Oriented Business Process Modelling, ECOOP, Brussels, 1998, available at http://www.ibissoft.se/ooworkshop.htm (task modelling, structured to generate scenarios)

[Alexander 2001: Alexander, Ian, *Visualising Requirements in UML*, article for Telelogic NewsByte, 2001, available at http://easyweb.easynet.co.uk/~iany/consultancy/reqts_in_uml/reqts_in_uml.htm (goal-directed use case modelling)

[Alexander 2002: Alexander, Ian, *Initial Industrial Experience of Misuse Cases in Trade-Off Analysis*, 6th IEEE International Symposium on Requirements Engineering, Essen, 11-13 September 2002 (application of misuse cases to conflicting industrial requirements)

[Scenario Plus 2002: website, http://www.scenarioplus.org.uk (free Use/Misuse Case toolkit for DOORS)

[Jacobson 1992: Jacobson, Ivar, et al: *Object-Oriented Software Engineering: A Use Case Driven Approach*, Addison-Wesley, 1992 (use case concept)

[Cockburn 2001: Alistair Cockburn, *Writing Effective Use Cases*, Addison-Wesley, 2001 (use case structure)

[van Lamsweerde 1998: van Lamsweerde, Axel and E. Letier, *Integrating Obstacles in Goal-Driven Requirements Engineering*, Proceedings ICSE'98 - 20th International Conference on Software Engineering, IEEE-ACM, Kyoto, 19-25 April 1998. (KAOS approach to goal-obstacle analysis)

[Anton & Potts 1998: Antón, Annie and Colin Potts, *The Use of Goals to Surface Requirements for Evolving Systems,* in Proceedings of the 20th International Conference on Software Engineering (ICSE`98), Kyoto, Japan, pp. 157-166, 19-25 April 1998. (goal-obstacle analysis)

[Sindre & Opdahl 2000]: Sindre, Guttorm and Andreas L. Opdahl, *Eliciting Security Requirements by Misuse Cases*, Proc. TOOLS Pacific 2000, pp 120-131, 20-23 November 2000 (misuse cases)

[Allenby & Kelly 2001]: Allenby, Karen and Tim Kelly, *Deriving Safety Requirements Using Scenarios*, Proceedings of the 5th International Symposium on Requirements Engineering, 27-31 August 2001, Toronto, Canada, 228-235 (failure cases for aircraft safety)

[McDermott & Fox 1999]: McDermott, John and Chris Fox, Using Abuse Case Models for Security Requirements Analysis, 15th Annual Computer Security Applications Conference, IEEE 1999, 55-66 (abuse cases for security requirements)

[Chung 2000]: L. Chung, B.A. Nixon, E. Yu, J. Mylopoulos
Non-Functional Requirements in Software Engineering, Kluwer, 2000 (i* method)