

THE NEGOTIATION ANALYSIS PATTERN

Haitham Hamza¹, and Mohamed E. Fayad²

¹*Computer Science and Engineering Dept., University of Nebraska-Lincoln
Lincoln, NE 68588, USA
hhamza@cse.unl.edu*

²*Computer Engineering Dept., College of Engineering, San José State University
One Washington Square, San José, CA 95192-0180
m.fayad@sjsu.edu*

ABSTRACT

Negotiation is a general concept that has many applications in various contexts. This paper introduces the Negotiation analysis pattern. This pattern aims to provide a model that analyzes the core concept of the negotiation notation independent of any application. In order to achieve this goal, Negotiation pattern is built based on the concepts of Stable Analysis Patterns we have introduced before in [2, and 9]. The core idea behind stable analysis patterns is to build the model of the pattern based on the software stability concepts introduced in [6, 7, and 8]. That is to construct the pattern in terms of its Enduring Business Themes (EBTs) [5], Business Objects (BOs), and Industrial Objects (IOs). By applying this approach to the negotiation problem, the resultant pattern will poses the stability feature, and, therefore, it can be used to model the negotiation problem in any application. The paper provides detailed documentation of the proposed Negotiation pattern.

CONTEXT

In our every day life, there are various situations where negotiation usually can place. For instance, buying or selling usually involves some sort of negotiation (e.g. buying or selling a home or a car). In software systems, negotiation appears frequently in the development of different applications. For instance, developing software for online auctions and shopping might involve the negotiation of the price and/or the negotiation of different product aspects.

More technically, negotiation is an essential part in the development of next generation Web-based devices and appliances. Today, devices that need to access the Web diverge greatly in their capabilities, making it highly desirable for the same resource to be available in several different representations (different languages for example). Negotiation algorithms play a fundamental role in aiding servers to decide which representation of a document a device should be given. In this case, the browser (or the client agent) will indicate its preferences by including a header in the request.

PROBLEM

Due to the wide range of the applicability of the negotiation concept, having a stable pattern that can model the basic aspects of a negotiation problem would make it easier to

THE NEGOTIATION ANALYSIS PATTERNS

build systems by reusing and extending this pattern. But the main problem is: How can we build a negotiation model that can be used to model the negotiation problem in any application?

FORCES

Negotiation pattern should resolve the following forces:

- Negotiation spans many applications that are completely different in their natures.
- The Negotiation process can take place between two or more persons, between persons and organizations, or between two non-human entities; in each case, the negotiator structure is completely different. How can we handle these different structures using a single model?
- The Negotiation entity can be an organization consisting of many persons, each having a role in the negotiation process. For instance, there can be one individual who is responsible for negotiating financial issues, another who is responsible for negotiating issues related to management, and so on. Therefore, our pattern should be flexible enough to handle different negotiator structures.
- Negotiation can be conducted through one or more media either simultaneously or consecutively; thus, the pattern should handle the use of multiple media.
- Negotiation can be performed on one or more affairs at the same time. For instance, negotiation that takes place in the buying and selling context usually involves more than one subject to be negotiated. For example, in buying a car, one can negotiate the price, the warranty, and so on. Therefore, the pattern should be flexible enough to handle such situations.
- The ultimate goal of any negotiation is to reach an agreement between the negotiators. However, the nature of this agreement varies tremendously from one application to another and from one context to another. An agreement that might be reached while negotiating a political conflict is completely different from that reached while buying a car. Therefore, the pattern should be able to handle these wide variations.

PATTERN STRUCTURE AND PARTICIPANTS

Figure 1 shows the object diagram of the *Negotiation* pattern. The pattern contains

THE NEGOTIATION ANALYSIS PATTERNS

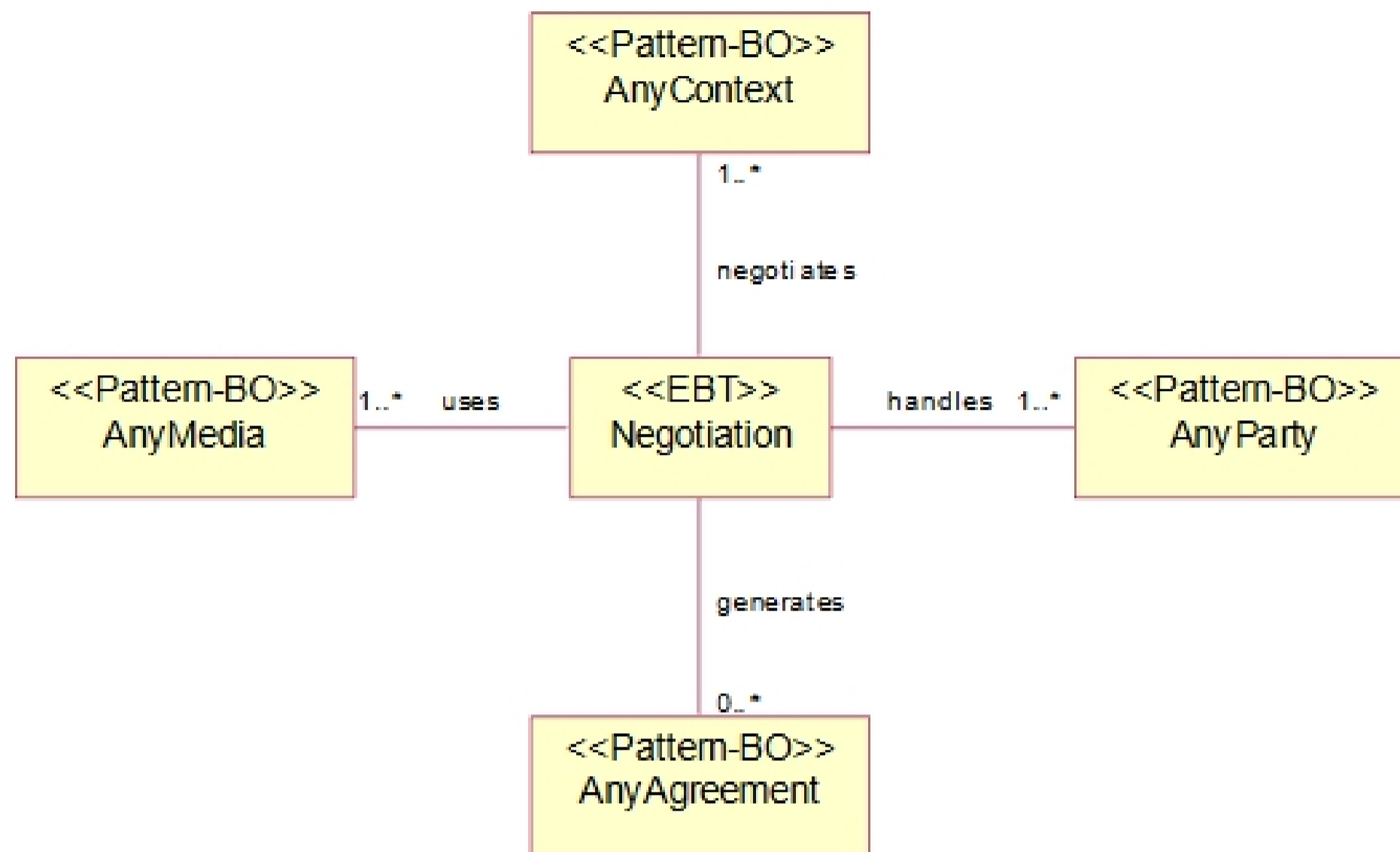


Figure 1. *Negotiation* pattern stable object model

As shown in Figure 1, the *Negotiation* pattern consists of the following participants:

Classes:

- *Negotiation*: Represents the negotiation process itself. This class contains the behaviors and attributes that regulate the actual negotiation process.

Patterns:

- *AnyAgreement*: Represents the result of the negotiation. The ultimate goal of any negotiation is to reach an agreement. Thus, this object presents a core element in any negotiation. It is important to note that in many cases negotiation ends with no agreement and thus it is considered to be failed (the seller of the car did not agree on the price proposed by the buyer and vice versa), however, in this case we expect that the agreement should provide this result by whatever mechanism. So one can view the agreement object as the result of the negotiation, which is not necessary a successful result.
- *AnyParty*: Represents the negotiation handlers. It models all the parties that are involved in the negotiation process. Party can be a person, organization, or a group with specific orientation. The pattern diagram and detailed pattern description is provided in [9].
- *AnyMedia*: Represents the media through which the negotiation will take place. For instance, one can negotiate the price of a good over the phone. Others might use an