

# A secure analysis pattern for handling legal cases

Eduardo B. Fernandez (\*), David L. la Red M. (\*\*), Jorge Forneron (\*\*\*), Valeria E. Uribe (\*\*), and Gisela Rodriguez G. (\*\*)

(\*) Dept. of Computer Science and Eng., Florida Atlantic University, Boca Raton, FL, USA

(\*\*) Dpto. de Informática, Facultad de Cs. Exactas, Universidad Nacional del Nordeste, Corrientes, Argentina

(\*\*\*) Dpto. de Informática, Facultad de Ciencias Aplicadas, Universidad Nacional de Pilar, Pilar, Paraguay

## Abstract

We present here a pattern of a type that we call a *Secure Semantic Analysis Pattern (SSAP)*. This is an analysis pattern that combines functional and security aspects. In particular, we present here a SSAP to handle legal cases. This pattern describes the handling of legal cases where a client is either suing another party (a plaintiff) or is being defended from a suit (a defendant). To describe SSAPs we have extended the POSA template with sections on possible attacks (the possible attacks in each action of a use case), needed policies (to prevent or mitigate the attacks), and secure structure (the class model of the solution with security constraints).

## 1. Introduction

We have proposed the use of Semantic Analysis Patterns (SAPs) to build conceptual models of applications [Fer00]. A SAP is a composite pattern that corresponds to a few fundamental use cases. Using SAPs is possible to build conceptual models in a simpler and more reliable way. We have also developed a methodology to build secure systems [Fer06a]. In this methodology we add instances of security patterns to the functional parts of the conceptual model to define security constraints at the application level. These constraints are then enforced by the lower architectural levels.

We present here a pattern of a type that we call a *Secure Semantic Analysis Pattern (SSAP)*. This is an analysis pattern where we consider a few fundamental use cases, we analyze the activities in these use cases, we consider possible attacks to them, and we define policies to prevent the attacks. This is the application of an idea proposed in [Fer06b] which emphasizes that the secure design of a system should be based on its expected types of attacks. Since the SAPs are used to build the conceptual model of an application, we have now a portion of a conceptual model where functional and security aspects are integrated from the start. In particular, we present here a SSAP to handle legal cases. To describe SSAPs we have extended the template of [Bus96] with sections on possible attacks (the possible attacks in each activity of a use case), needed policies (to prevent or mitigate the attacks), and secure structure (the class model of the solution with security constraints). SSAPs follow the current tendency in security research of integrating business functions with security aspects from the beginning of the development life cycle [Nag05, Sch06a].

Section 2 describes an example of an SSAP, a pattern for the Secure Handling of Legal Cases. A glossary at the end of the paper defines basic law terms.

## 2. Secure Handling of Legal Cases

This pattern describes the handling of legal cases where a client is either suing another party (a plaintiff) or is being defended from a suit (a defendant). The pattern includes the necessary policies to stop or mitigate the expected attacks.

### 2.1 Example

The SueThem law firm is having trouble staying in business. It keeps some documents in electronic form and others in paper. Documents are hard to find and get easily accessed by unauthorized persons. It is hard for the company to keep track of their customers and to know how much it should charge them. The conduction of cases is disorganized, which leads to losing cases because of lack of preparation.

### 2.2 Context

A legal firm sues parties (persons, organizations, or groups) on behalf of their clients; it can also defend those clients when they are sued. We call a *legal case* the sequence of actions (process) needed to pursue a suit until its completion. The standard legal system of most countries allows parties to sue other parties. There are different types of lawsuits but they are not of interest here. Interactions between the people involved can be in person, by telephone, by regular mail, or by email. Law firms are commercial entities and must compete with other law firms for clients.

### 2.3 Problem

A law suit or defense implies a sequence of actions and generates many documents of several types. If the firm doesn't organize properly these actions and the corresponding documents, it will have problems in conducting the suit or defense, which will result in unnecessary expenses and in a higher possibility of losing the case. Because the information handed in a case is very sensitive, there is motivation to misuse it. We need to consider possible attacks and take measures to avoid them. We consider here the main use cases in this process: Handle Legal Case (for a plaintiff), Handle Legal Case (for a defendant), and its auxiliary use cases Keep Track of Costs, Research Case, and Billing. Figure 1 shows the actors involved in these use cases. 'Other' represents here people involved in the case such as witnesses or experts. There are other related use cases which are left out for simplicity and to make the pattern more reusable.

The solution to this problem is affected by the following forces:

- The sequence of activities in a case is usually unpredictable. Depositions, witness court appearances, lawyer briefs to the court might be required in any sequence depending on the course of the case.
- Complex cases may require several lawyers with the assistance of some secretaries. The actual number of these people might be hard to predict.
- In addition to the defendant and the plaintiff (and their respective opponents) we may need witnesses, experts, and other people. Who they are and when they are needed depends on the case.
- The total effort and duration of a case is variable and we need to keep track of expenses, time used, supplies, etc., so we can bill our clients.

- Handling cases require searching for precedents (similar cases). To do research for cases, lawyers and secretaries make use of libraries and the Internet and may download many documents.
- The information about customers, billing, assignment of lawyers, and other aspects related to a current case must be accessible only to authorized persons.
- Legal documents can only be created by authorized persons and their use (reading or modification) should also be controlled.
- Government regulations apply to law firms and their information must be easily auditable.

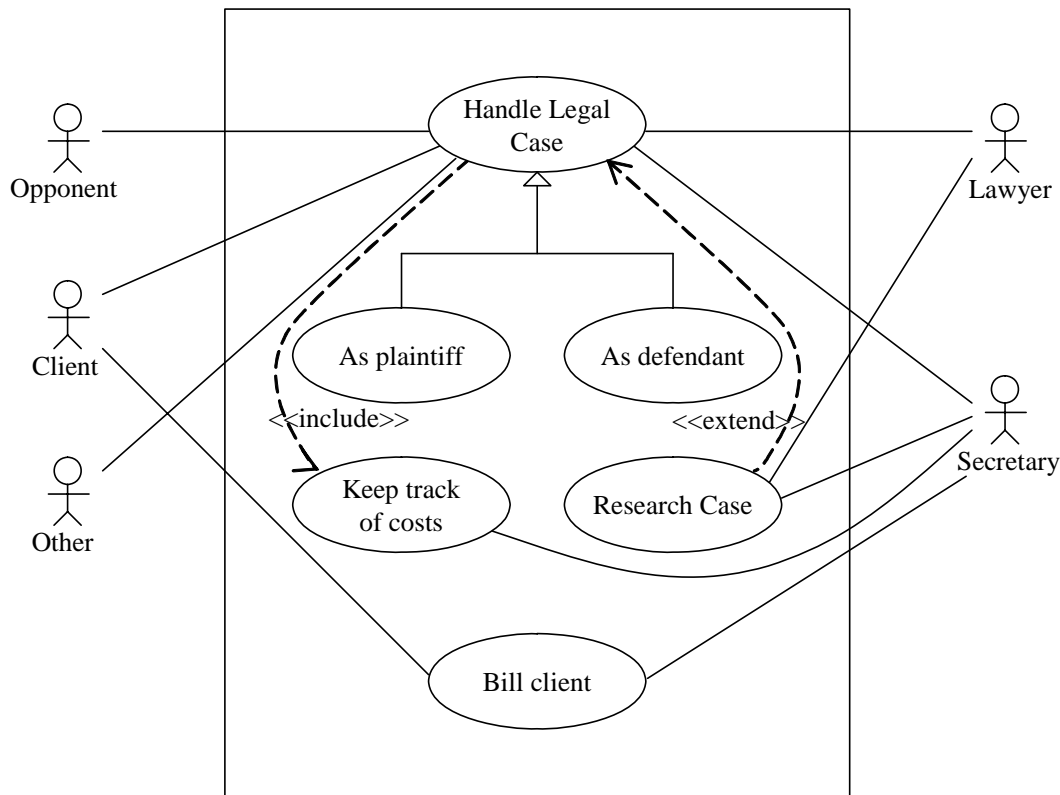


Figure 1. Use cases for handling legal cases

## 2.4 Possible attacks

Figure 2 shows an activity diagram for the sequence for handling a case followed by billing, tracking of costs, and related case research. Following the approach of [Fer06b], to analyze the possible attacks we consider each activity in the diagram of Figure 2 and see how it can be subverted by the attacker. In this diagram External People indicates either the opponent or other people involved in the case. The possible attacks or threats are then:

- A1 In the 'start case' activity, the client or the responsible lawyer might be impostors.
- A2. A lawyer might create a false contract
- A3 The client or the external people might give a false deposition.

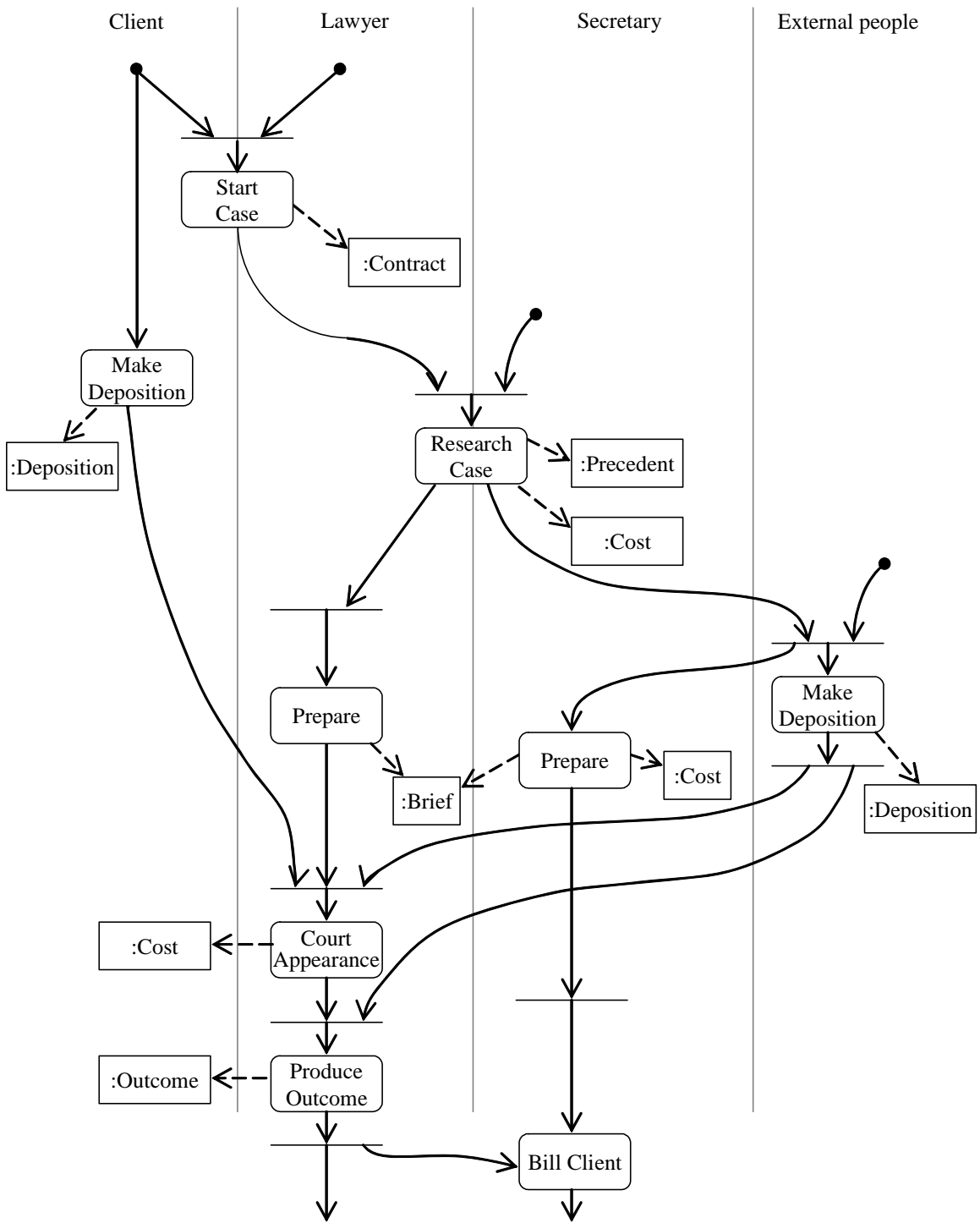


Figure 2. Activity diagram of a case handling

A4 A lawyer may change a deposition.

A5 A lawyer or a secretary may produce intentionally incorrect precedents, briefs, or costs.

A6 A secretary may produce an increased or decreased bill.

A7 A lawyer may change some aspects of the outcome to collect a higher fee.

A8 A lawyer can disseminate client or case information for monetary gain.

A9 An external attacker may read/change case information.

## 2.5 Solution

Because the handling of cases is rather unpredictable and we use a variety of knowledge experts in its handling, this problem can be modeled as a Blackboard pattern [Bus96]. The case itself becomes a blackboard and the experts providing knowledge to the case are the lawyers, witnesses, or experts. The control is based on the status of the case and is embodied in the scheduling of activities.

### *Structure*

Figure 3 shows a class diagram of the conceptual model for the functional aspects of this pattern. Class **Case** represent the case itself (in the role of Blackboard), and it includes as components classes **Cost** (describes accrued costs), **CaseDocument**, **Outcome** (the result of the case), and **Scheduling** (the control role of the Blackboard). A **Client** is responsible for a case, and with each case there are some associated **ExternalPeople** (opponents, witnesses, experts). A CaseDocument can be a **Contract**, a **Precedent**, a **Brief**, or a **Deposition**. Lawyers and Secretaries are **Employees** of the **Law Firm**. Lawyers and Secretaries can be assigned to cases (we assume this assignment has been done beforehand). A Secretary in the case keeps track of Costs. A Lawyer in the case is responsible for the general conduction of the case, including scheduling.

### *Dynamics*

Figure 4 shows a sequence diagram describing some typical steps for the use case Handle Legal Case as Plaintiff. The Client starts the case with the responsible lawyer. This lawyer creates an instance of a case and later does some research for it. He assigns an assistant lawyer to prepare a brief for the court and schedules the client to make a deposition.

### *Needed policies*

The attacks identified earlier mean that we need the following policies to avoid or mitigate them:

A1 Mutual authentication, to avoid impostors.

A2 Authorization to restrict only lawyers to create contracts, and logging to record possible illegal actions from a lawyer.

A3 Logging, to keep records for future auditing that could detect false depositions.

A4 Authorization and document protection against change

A5 Authorization and logging, to restrict who can perform these actions and to keep records for future auditing.

A6 Logging, to record suspicious actions of a secretary.

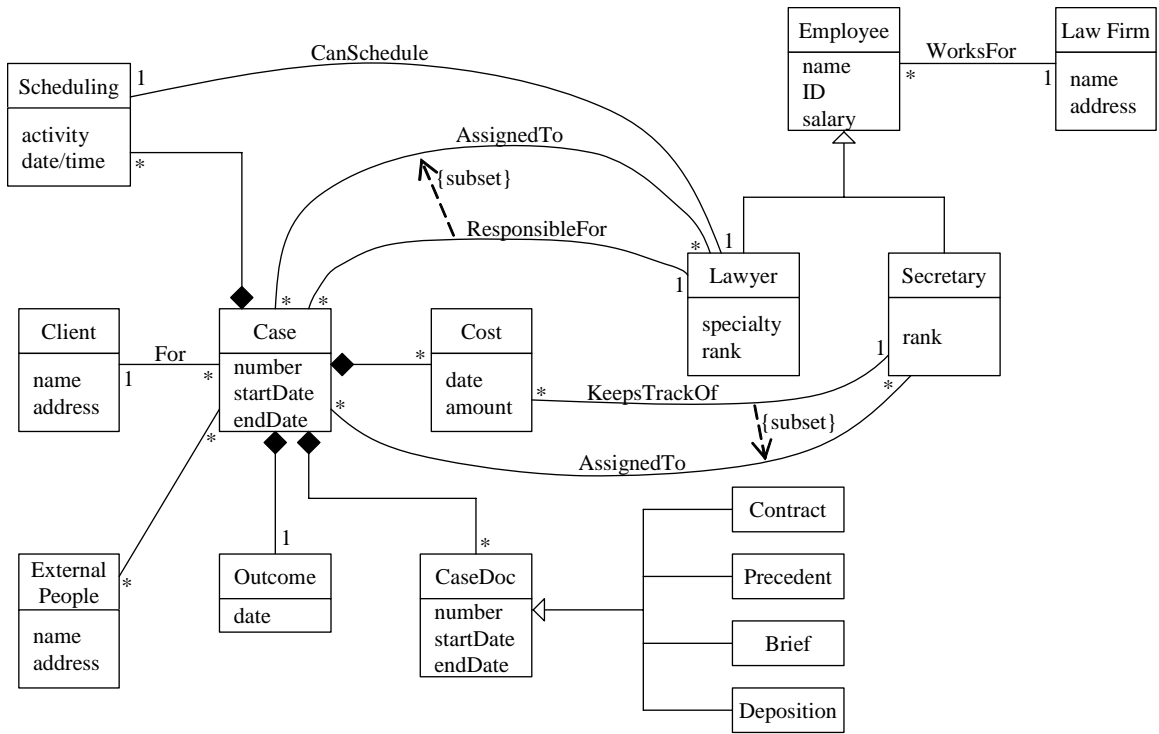


Figure 3. Class diagram for the Handle Legal Cases pattern.

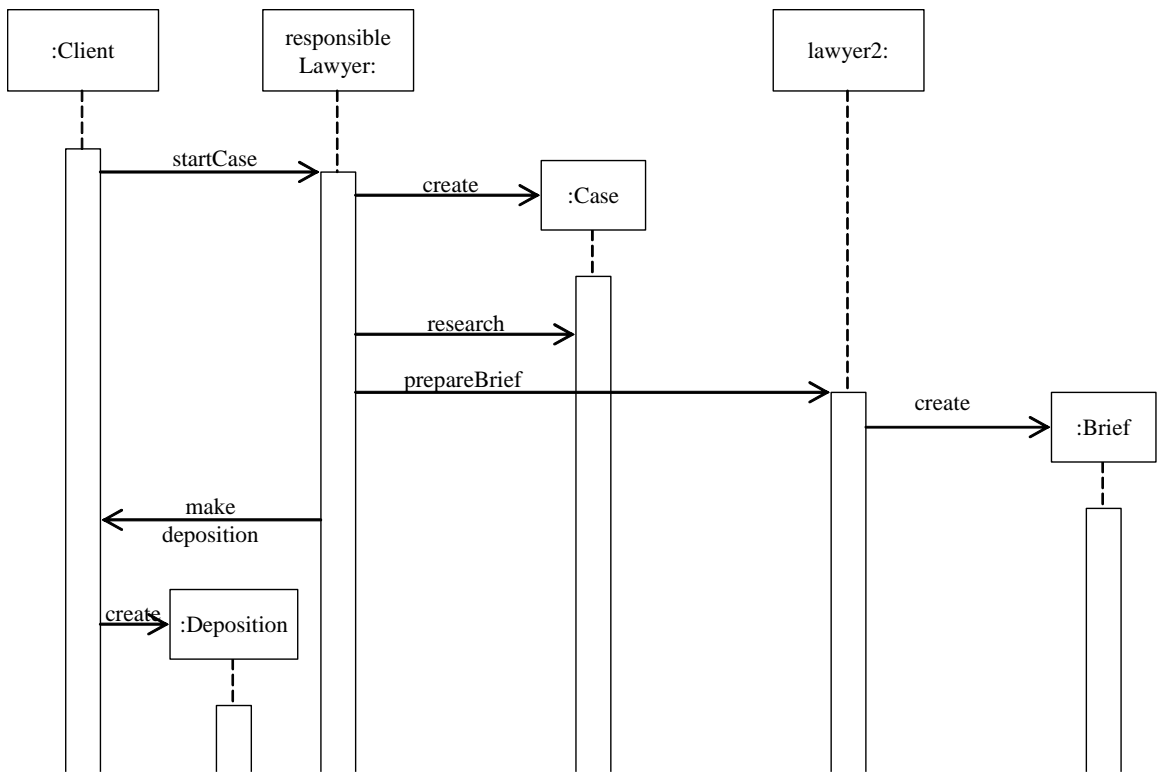


Figure 4 Sequence diagram for use case 'Handle Case'.

- A7 Separation of duty. Two lawyers must concur on the fees to be charged.
- A8 Logging, to record possible illegal actions of lawyers.
- A9 Authorization and access control to stop external attacks.

**Secure structure**

Figure 5 shows the conceptual model of Figure 3 with the addition of instances of Authentication, Authorization, and Logging patterns to realize the identified policies. We assume that the authorization policies use Role-Based Access Control (RBAC). Both the responsible lawyer (who interacts with the client), and the client must have information to authenticate each other (**Authenticator**). The **CaseLog** records accesses to the case data. We also need an instance of the Reference Monitor, not show here for simplicity (see [Fer06b]).

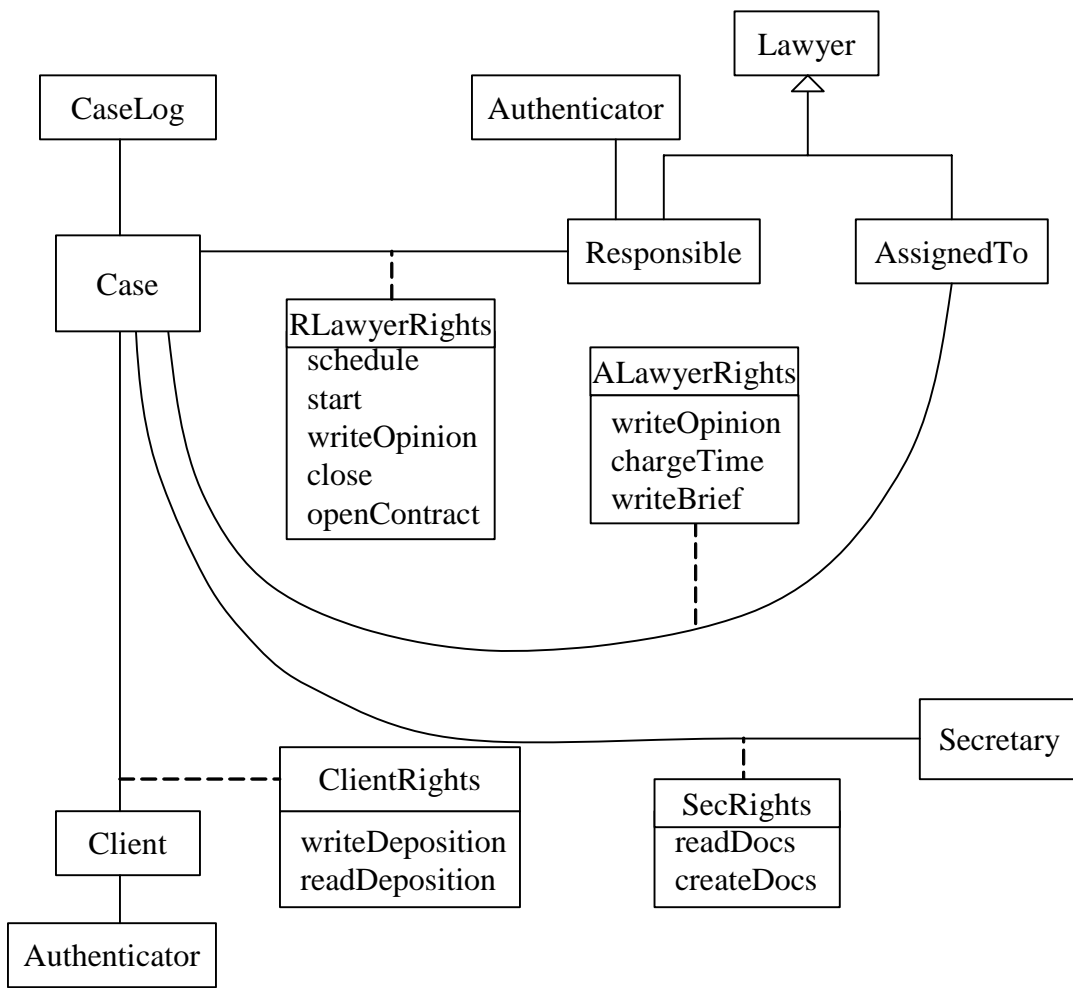


Figure 5. Security additions to the class diagram.

### **Example resolved**

The SueThem law firm has now a systematic structure to conduct its cases. All its documents are reflected in the conceptual model and can be easily retrieved and audited. The company can now keep track of the costs associated with a case. Documents and other case information can be protected from illegal access.

### **Consequences**

This pattern has the following advantages:

- The Blackboard structure accommodates well unpredictable sequences of activities.
- We can assign lawyers and secretaries dynamically depending on the course of the case.
- The model includes knowledge sources that can be the client, the opponent, witnesses, expert witnesses, and other people.
- It is possible to track the current costs of the case.
- Applying legal regulations to the company is easy because all documents are described by classes with controlled access and we keep a log of accesses.
- Searching for precedents (similar cases) can be done as part of the case handling, we can store this information for future use, and we can associate it to the different stages of the case.
- The RBAC structure enforces authorized access to the information and employees can make sure that they are talking to the person they intend.
- Cryptographic methods can be added to prevent document modification, e.g. hashing [Gol06].

Liabilities include:

- The order in which some activities are performed has an effect in the outcome but the lawyers must decide on the scheduling and the pattern does not help here.
- We might not be able to find all possible attacks, which could allow some attacks to still happen.
- The actual implementation may allow new types of attacks.

### **Effect on security:**

- We can define precise role rights, e.g. an expert can only add to the information, not change it, a lawyer can decide on the next step, bring new witnesses, but cannot change depositions.
- A designer building a system of this type can produce software that performs its functions and is at the same time reasonably secure.

### **Known uses**

Many large law firms follow a similar structure.

### **See also**

- The *Blackboard* pattern [Bus96] is the basis for the central function of the case.
- The client and the external people can be described by a *Party* pattern to indicate that they can be individuals or organizations [Fow97].

- Assignment of lawyers and secretaries uses the *Resource Assignment* pattern [Fer05].
- The rights structure follows an RBAC pattern [Sch06b].
- Authentication is performed by means of instances of the Authenticator pattern [Sch06b].

## Acknowledgements

We thank our shepherd, Jorge Ortega Arjona, who has provided valuable suggestions that have clearly improved this paper.

## References

- [Bus96] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal. *Pattern Oriented Software Architecture: A System of Patterns, Volume 1*, Wiley, 1996.
- [Fer00] E.B. Fernandez and X. Yuan, "Semantic analysis patterns", *Procs. of 19th Int. Conf. on Conceptual Modeling*, ER2000, 183-195. Also available from: <http://www.cse.fau.edu/~ed/SAPpaper2.pdf>
- [Fer05] E.B.Fernandez, T. Sorgente, and M. VanHilst, "Constrained Resource Assignment Description Pattern". *Proceedings of the Nordic Conference on Pattern Languages of Programs, Viking PLoP 2005*, Otaniemi, Finland, 23-25 September 2005.
- [Fer06a] E. B. Fernandez, M.M. Larrondo-Petrie, T. Sorgente, and M. VanHilst, "A methodology to develop secure systems using patterns", Chapter 5 in *Integrating security and software engineering: Advances and future vision*, H. Mouratidis and P. Giorgini (Eds.), IDEA Press, 2006, 107-126.
- [Fer06b] E. B. Fernandez, M. VanHilst, M. M. Larrondo Petrie, S. Huang, "Defining Security Requirements through Misuse Actions", in *Advanced Software Engineering: Expanding the Frontiers of Software Technology*, S. F. Ochoa and G.-C. Roman (Eds.), International Federation for Information Processing, Springer, 2006, 123-137.
- [Fow97] M. Fowler, *Analysis Patterns-Reusable Object Models*, Addison-Wesley, 1997
- [Gol06] D. Gollmann, *Computer security (2<sup>nd</sup> Ed.)*, Wiley, 2006.
- [Nag05] N. Nagaratnam, A. Nadalin, M. Hondo, M. McIntosh, and P. Austel, "Business-driven application security: From modeling to managing secure applications", *IBM Systems Journal*, Vol. 44, No 4, 2005, 847-867.
- [Sch06a] A. Schaad, "Security in Enterprise Resource Planning systems and Service-Oriented architectures", *Procs. of SACMAT'06*, ACM, June 2006, 69-70.
- [Sch06b] M. Schumacher, E.B.Fernandez, D. Hybertson, F. Buschmann, and P. Sommerlad, *Security Patterns: Integrating security and systems engineering*, Wiley 2006.

## **Appendix.      Glossary of legal terms**

**Brief**--a formal document that sets forth the main contentions with supporting statements or evidence.

**Contract**--a binding, legally enforceable agreement between two or more parties.

**Defendant**--a person required to make answer in a legal action or suit.

**Deposition**—a testimony taken down in writing under oath.

**Expert**—a person having or displaying special skill or knowledge derived from training or experience.

**Opponent**--one that takes an opposite position (as in a debate, contest, or conflict).

**Plaintiff**--a person who brings a legal action.

**Precedent**--something done or said that may serve as an example or rule to authorize or justify a subsequent act of the same or an analogous kind.

**Suit**--an action or process in a court for the recovery of a right or claim.

**Witness**--one who testifies in a cause or before a judicial tribunal.