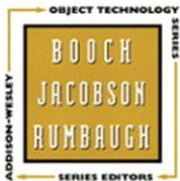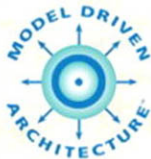# ENTERPRISE PATTERNS AND MDA

BUILDING BETTER SOFTWARE WITH ARCHETYPE PATTERNS AND UML

JIM ARLOW

ILA NEUSTADT

Foreword by **Richard Mark Soley**

MODEL DRIVEN ARCHITECTURE

Praise for *Enterprise Patterns and MDA*

"The burgeoning field of Model Driven Architecture tools and worldwide support for the Unified Modeling Language are finally being met with high-quality books that explain standard modeling techniques in a way any developer can follow. This book meets an urgent need squarely and clearly, and explains with copious examples a powerful approach to building usable (and reusable!) assets and applications. Every enterprise developer needs this book."

>—Richard Mark Soley, Ph.D.
>Chairman & CEO
>Object Management Group, Inc.

"I've never seen a system of business patterns as detailed as this one. The completeness that Arlow and Neustadt provide in these patterns is impressive. The explanations for why the patterns are formed the way they are and how they're interconnected are incredibly thorough. The patterns presented here have the potential to impact business applications in the same way the 'Gang of Four' patterns have impacted general software development."

>—Steve Vinos ki
>Chief Engineer of Product Innovation
>IONA Technologies

"[*Enterprise Patterns and MDA* is a] detailed, yet very readable, guide to designing business applications using reusable model components and Model Driven Architecture. It deserves a place on every application designer's desk."

>—Andrew Watson
>Vice President and Technical Director
>Object Management Group, Inc.

"Design patterns are generally acknowledged as an effective approach to developing robust and highly reusable software. Now that Model Driven Architecture is raising software design to ever-higher levels of abstraction, it is only natural that pattern concepts should find application in advanced modeling techniques. With this book, Arlow and Neustadt have greatly advanced the state of the art of MDA by defining both a theory and a methodology for applying the concept of Archetype Patterns to business software modeling."

>—John Poole
>Distinguished Software Engineer
>Hyperion Solutions Corporation

System Log On

identify party

authenticate party

tries = the number of attempts at authentication

maxTries = the number of permissible attempts at authentication

[(!success) AND (tries <= maxTries)]

[(!success) AND (tries > maxTries)]

handle security violation
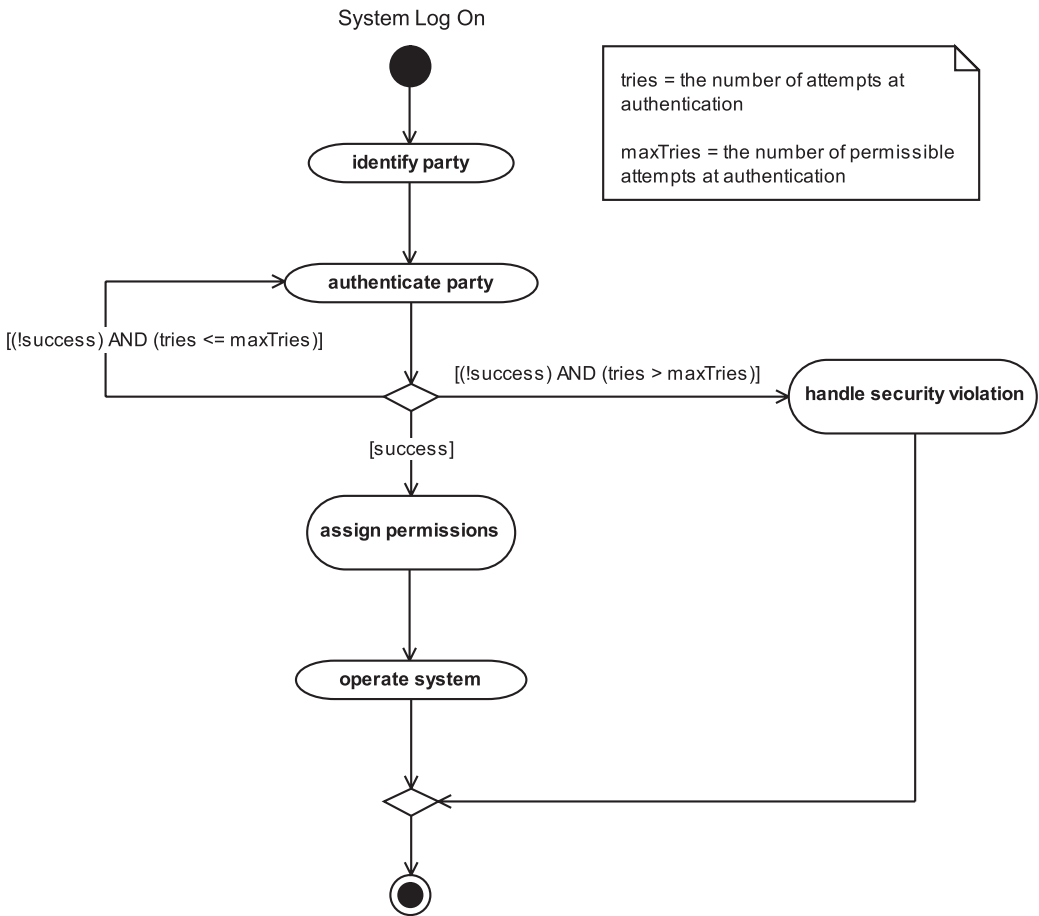
[success]

assign permissions

operate system

Figure 4.6

The degree of security you require will determine which of these options is appropriate.

Option 1 is the most secure if the `Party` is required to keep the user string secret, but it is predicated on the `Party` remembering this string.

Option 2 tries to take away the requirement to remember a user string, by attempting identification based on common personal information such as name and postal code, or perhaps a passport number. This can be a less secure approach because common personal information may be readily available to anyone, not just the `Party` in question. Also, depending on the personal information requested, there may be more than one `Party` instance returned.

Option 3, automatically obtaining the `Party`'s ID from his or her machine, is always attractive in selling systems because it provides the lowest barrier to entry of the system. But from the user's perspective it is the least secure. Depending on how `authenticate party` is implemented, this approach may allow other people to buy goods or access services on the user's account without his or her knowledge or permission.

authenticate party

Authentication is about establishing that a `Party` is who, or what, the `Party` claims to be. Usually, this is done by exchanging a secret token such as a password, or even a security certificate, across a secure channel. On-line selling systems may give the `Party` the option to store this password in a persistent form on his or her machine so that the system can access it automatically. Again, this is the least secure option—you should *always* give your users advice about the security implications of this, and *always* give them the option to turn the feature off.

handle security violation

Most systems allow a maximum number of authentication attempts, perhaps in a fixed time period, and then treat failure as a security violation. This violation may result in actions with varying degrees of severity.

1. Do nothing.
2. Log the violation.
3. Log the violation and immediately notify the security administrator.
4. Log the violation, immediately notify the security administrator, and suspend the user's account (in the case of a physical token, such as a card in an ATM, the token may be confiscated or otherwise rendered invalid).

Your least extreme option is to do nothing—but this is rarely, if ever, appropriate! The least extreme *realistic* option is simply to log the violation for inspection by a security administrator. This is not a very secure approach because it may leave your system open to brute-force cracking if the administrator doesn't handle the violation in a timely way. Options 3 and 4 above provide more secure approaches.

assign permissions

This activity assigns to the `Party` permissions to access various system resources.

`operate system`
The `Party` operates the system in some way.

4.10 Address

The `Address` archetype represents information that can be used to contact a `Party`.

This is illustrated in Figure 4.7 where you can see that:

- A `Party` may have zero or more associated `Addresses`.
- There are four kinds of `Addresses`—we will discuss each of these in the next four subsections. Business rules should be applied to determine which `Address` to use in a particular situation.
- Each time an `Address` is associated with a `Party`, there is an associated `AddressProperties` object (defined below). This object should at least specify the `use` of the `Address` (i.e., limit the context in which the Address is applicable—business, home, out-of-hours contact, emergency contact, and so on). You can also employ `use` to identify whether an Address is a preferred address for the `Party`. Other properties may be added to `AddressProperties` according to your business requirements.

The `AddressProperties` archetype specifies information about an `Address` assigned to a specific `Party`.

4.10.1 GeographicAddress

The `GeographicAddress` archetype represents a geographic location at which a `Party` may be contacted. It is a postal address for the `Party`.

The semantics of the attributes of `GeographicAddress` (see Figure 4.7) are summarized in Table 4.2 on page 134.
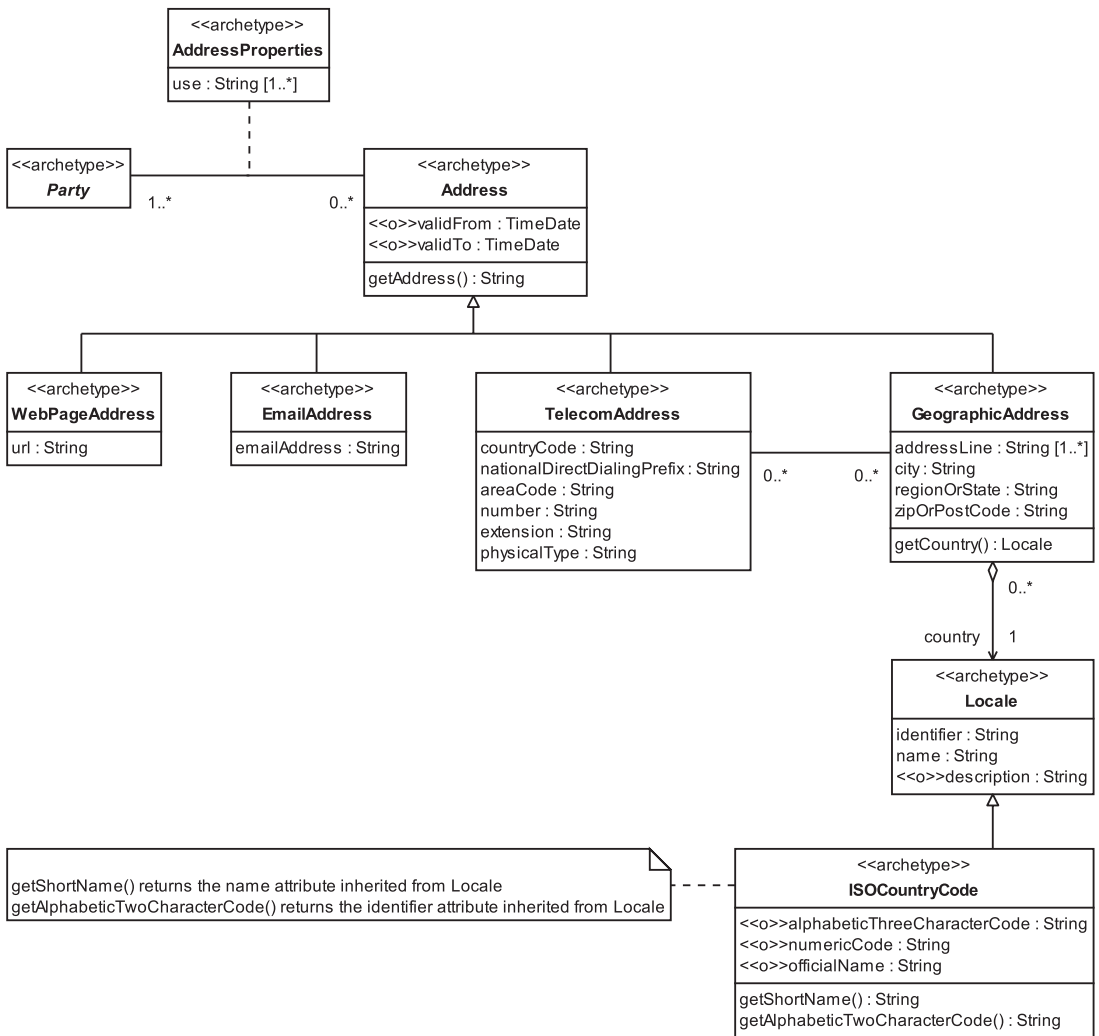
Figure 4.7

Each `GeographicAddress` has exactly one `Locale` (see Section 11.6). This is often an `ISOCountryCode` (also discussed in Section 11.6) that represents the alphabetic and numeric country codes and English language country names defined in ISO 3166.

ISO 3166 is an international standard published by the International Organization for Standardization (ISO). You can find out about ISO standards at

Table 4.2

`GeographicAddress` **archetype**	
Attribute	**Semantics**
`addressLine[1..*]`	One or more address lines
	The first line of an address is often a building name, a building number and street, or a postal box number; the other address lines are for location information that is best placed on its own line
`city`	The name of a city or town
`regionOrState`	The name of a geographical region within a country—in America this would be a state; in England, a county; and in Switzerland, a canton
`zipOrPostCode`	A code used to identify a geographic area

the ISO Web site, www.iso.org. This site allows you to purchase ISO standards documentation and has complete and up-to-date lists of country codes and country names available for download, in HTML and semicolon-delimited text formats.

4.10.2 TelecomAddress

The `TelecomAddress` archetype represents a number that can contact a telephone, mobile phone, fax, pager, or other telephonic device.

In the future there may be types of telephonic devices that we don't know about yet. For example, there are already home automation systems that allow you to phone up the controller and instruct or interrogate it using your telephone's touch tone keypad.

The International Telecommunication Union (www.itu.int) provides standards for `TelecomAddresses`. Each address is made up of the parts shown in Figure 4.8.

National Direct Dialing prefix (NDD)

+44 (0)208 123 4567 ext. 789

Replace with the
International
Direct Dialing
prefix (IDD) for
your country
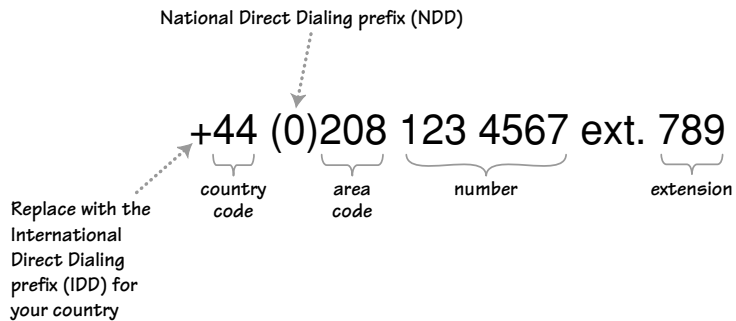
country
code

area
code

number

extension

Figure 4.8

Each part of the number, except the IDD, is represented by an attribute in the TelecomAddress archetype as described in Table 4.3. The IDD is not part of the TelecomAddress because it is the number you have to *add* to the Telecom-Address to access international direct dialing. This number depends on which country you are in when you try to direct dial out, so it is usually represented by a plus sign. For example, in the United Kingdom the IDD is 00, in the United States it is 011, and in France and several other countries it varies depending on which telecom provider you want to use for your international call. (See www.kropla.com for detailed listings of country codes, IDDs, and NDDs.)

Table 4.3

TelecomAddress **archetype**	
Attribute	**Semantics**
countryCode	The number you must use to direct dial a particular country—this is *not* the same as the ISO 3166 country code, which is an alpha code
nationalDirectDialingPrefix	The prefix you use to make a call within a country between different cities or areas—generally dropped when dialing from outside the country (exceptions include Italy, where the national-DirectDialingPrefix is included when dialing from outside; and Spain, which does not currently use a nationalDirectDialingPrefix)

Table continued on next page