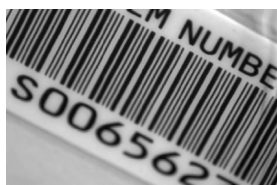


5 Patterns

5.1 Identification Pattern



Barcode is a machine readable strip for automatic identification of items, by means of printed bars of different widths

Context

People refer to real or imaginary things by their names. We name things to identify them, so we can refer to them by their names and not just point to them and say "this!". By naming, we give things identities, but in real life they are not often unique. Many things have more than one name, and sometimes a single name can refer to different things, which is fine as long as everyone who uses that name knows what thing it refers to. In business, people use serial numbers, production numbers, civil registration numbers, and names.

Problem

How do we specify the identity of things represented by REA entities?

Forces

The solution needs to balance the following forces:

- An identity is a given feature; it is not an intrinsic part of the objects and things. Therefore, an REA application model must specify whether there is a business reason requiring REA entities to have a distinct identity, and how that identity is modeled. We could omit modeling identity of an entity, but then we could distinguish different instances of this entity only by the values of their attributes.
- Users of business applications do not necessarily require that all REA entities have an explicit identifier. For example, users of business applications might not be interested in managing the identifiers of sales order lines.
- Some identifiers are unique in the universe, such as the GUID (Global Unique Identifier); some are not unique, such as the first name and last name of a person. Some identifiers are unique within a certain group, such as a serial number, which is unique within the group of entities that belong to the same number series.
- There are specific rules on how to construct identifiers. For example, the ISBN (International Standard Book Number) or the numbers of major credit cards are constructed in a way that enables verifying, using a simple calculation algorithm, whether the number is valid.

Solution

The *Identification Aspect Pattern* can be used in situations in which application developers want to specify the identity of REA entities.

In the REA application model, the *Identification Aspect* consists of two elements. The *Identifier* element represents the name or number of an REA entity. The *Identifier Setup* element specifies the rules for creating the Identifiers.

The *Identifier Setup* is often configured on *group* of REA entities that share the same rules for creating identifiers, for example, on a group that belongs to the same number series. The *Identifier* can be configured on any REA entity that needs to be identifiable, including the groups. As not all REA entities are parts of some group, the *Identifier Setup* is often omitted from the model, or is implicit in a software application, for example, as a system table.

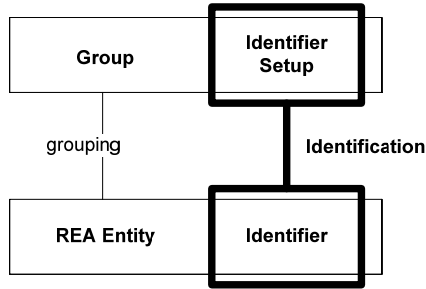


Fig. 111. Identification aspect in the application model

Design of the Identification Pattern

The *aspect type level* encapsulates the business logic of the aspect and configuration parameters, which can be set by application developers. At the aspect type level, the *Identification Type* defines the *Name* of the type of identification, as well as other attributes. *AutoNumber* is a Boolean function that can be set on or off to indicate whether the *Identifier* can be automatically generated by the identification aspect or not; automatically generated number is often referred to as a number series. *Unique* is a Boolean function that can be set on or off to indicate whether or not the *Identifier* is required to be unique at runtime. *Mandatory* is a Boolean function that can be set on or off to indicate if the *Identifier* must be defined at runtime or can be undefined.

The *Identification Type Aspect* has two elements, *Identifier Type* and *Identifier Setup Type*. These elements contain business logic for interpreting the *ID rules*, and logic for creating and validating *Identifiers*. They do not have any configuration parameters; just serve as metadata for the *Identifier* and the *Identifier Type* at the application level.

The rules for creating new *Identifiers* can vary from simple series with linear increments to rules that allow for validity checks of the identification strings, such as credit card numbers. Legislation in some countries requires that numbers of some business documents consecutive, without gaps, which imposes an extra requirement on how the number is constructed. If an REA entity has been created by omission and deleted after another REA entity of the same series has been created, the *ID Rule* must be able to identify the gap in the series and reuse the number of the deleted document.

The *application model level* specifies the runtime attributes that can be set by the users of the business application, or automatically. At the appli-

cation model level, the *Identifier* element is configured on the REA entity that should have some form of identity. The *Identifier* contains the *ID String*, which provides an identity to each REA entity instance.

The *Identifier Setup* is usually configured on a group of REA entities that share the same ID rule for creating or validating an Identifier. The *ID Rule* determines how the identification strings are created (users of business applications often use combinations of letters and numbers). The *ID Rule* can also be used for validating the identification strings entered manually by the users of the business application. If the *Identification Type* aspect is an *AutoNumber*, the *Identifier Type* also has an attribute *Last ID*, which defines the last used identification string in the series.

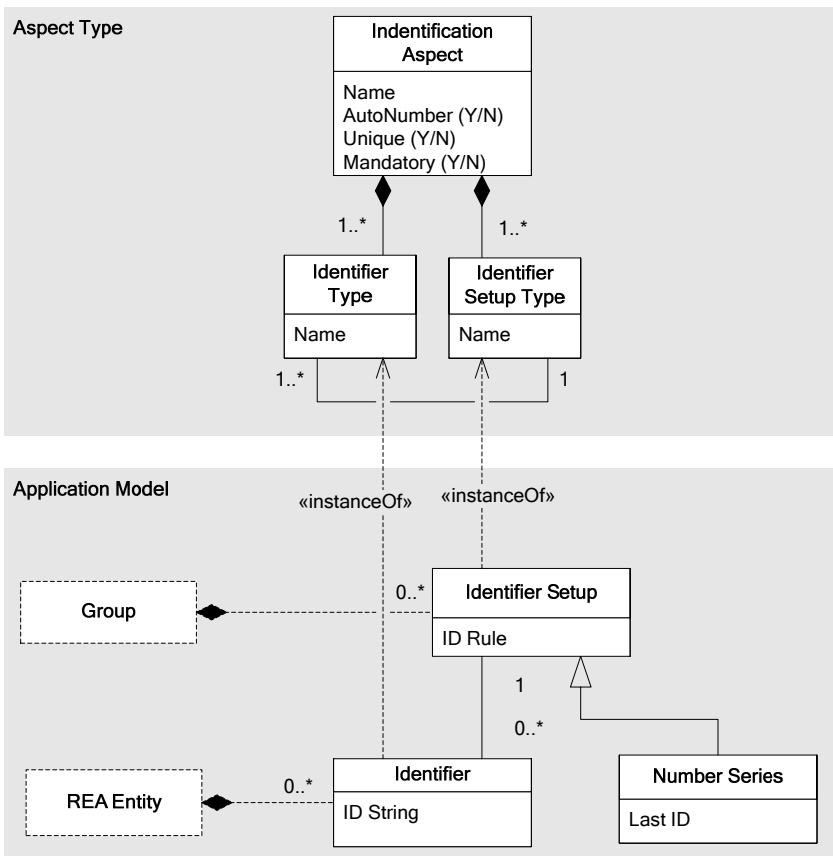


Fig. 112. Design of the identification pattern

Examples

The *Social Security Number (SSN)* of an employee is an identification that is not an *auto-number*, is *unique*, and is *not mandatory*. The *Identifier Setup* has the name *SSN Numbering Scheme*, and contains an *ID Rule* that determines how the social security number is constructed or verified. The *Identifier* has the name *Social Security Number*, and its *ID String* at run-time contains the social security number.

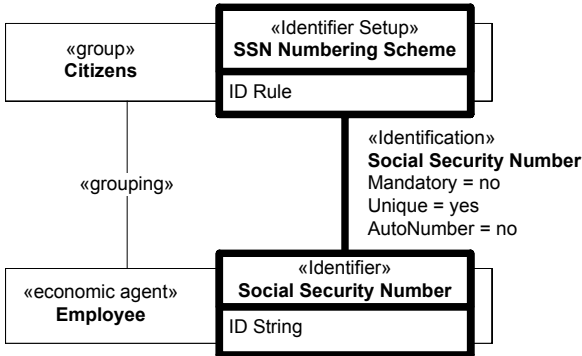


Fig. 113. Social Security Number

Sales Order Number is an identification that is an auto-number, is unique, and is mandatory. As the *Sales Order Number* is an auto-number, the *Identifier Setup* element contains the attribute *Last ID*.

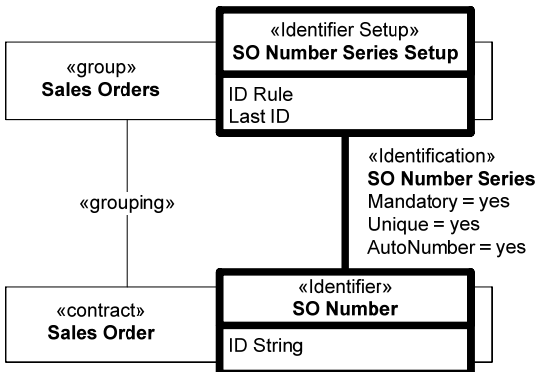


Fig. 114. Sales order number

Product Serial Number is an identification that is an *auto-number*, is *unique*, and is *mandatory*. The configuration of *Product Serial Number* is similar to that of *Sales Order Number* in Fig. 114; *Identifier* is configured on the economic resource *Product*, and *Identifier Setup* is configured on a group of *Products* that belong to the same series.

Employee Name is an identification that is not an *auto-number*, is not *unique*, but is *mandatory*. *First name*, *middle name*, *last name* and *nick-name* share the same *ID Rule* specified by *Employee Name Setup*.

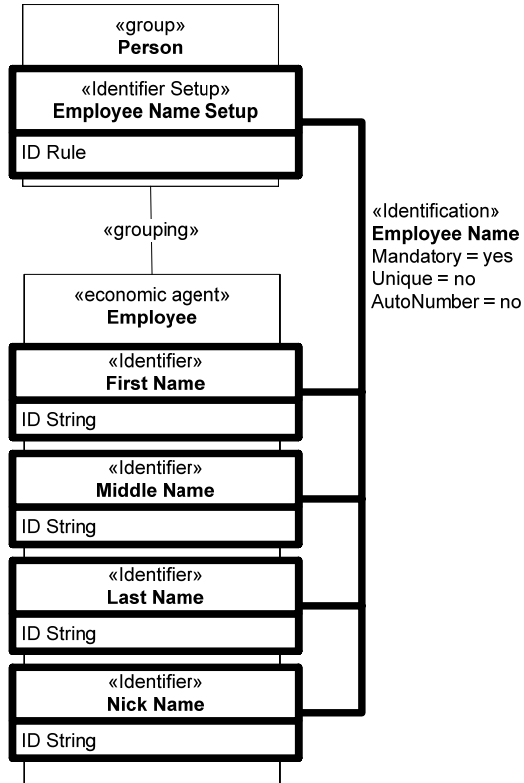


Fig. 115. Employee name

Resulting Context

Sometimes, users of business applications use *phone number*, *e-mail address*, or *Internet address* as identifiers of their trading partners. These numbers and addresses have multiple and different semantics. Phone num-

ber can also be used as a contact address, e-mail address as a contact address and destination location (for sending electronic documents and products), and Internet address as a description of the trading partner. In such cases, different aspects will contain or refer to the same data (both identification and notification will contain or refer to the same phone number).

There are several international standards specifying Identification Strings and ID rules for economic resources and economic agents in various lines of business. Examples are European Article Numbering (EAN) for industrial products, International Standard Book Number (ISBN) for books, International Standard Serial Number (ISSN) for periodicals, and International Standard Music Number (ISMN) for printed music publications. For companies, the Data Universal Numbering System (DUNS) is used. References to these standards can be found, for example, in (Arlow, Neustadt 2003).



<http://www.springer.com/978-3-540-30154-7>

Model-Driven Design Using Business Patterns

Hruby, P.

2006, XVI, 368 p., Hardcover

ISBN: 978-3-540-30154-7