Perspectives® as Actions of Roles in Contexts

A new approach to modeling Enterprise Information Systems

drs. ing. Cor Baars, drs. Joop Ringelberg, Research Department Perspect-IT

Summary

We propose Perspectives, a Multi Perspective, Multi Resolution (MPMR) (ref) modeling approach for Enterprise Information Systems that is easy to use for both Business- and Information Analysts. The goal of Perspectives is to simplify Information Modeling, create early prototypes and thereby improve Business – IT Alignment.
The central notion of Perspectives is summarized as: “Perspectives are the Actions of Roles in a Context” or: “The Perspective an Agent has on a Context is the set of Actions the Agent can perform within that Context”. Actions in this context refer to Business Actions that are decomposed into appropriate Information Actions such as creating, consulting and changing information. Modeling in Perspectives is therefore based on determining the Contexts, Roles and Actions in an application domain.

After some words on motivation and background, we introduce the Perspectives Diagram Language (PDL) and use it to show several Perspectives Design Patterns. These patterns are used in a motivating example in the education domain. Early experiences with PDL modeling have resulted in guidelines for the Perspectives Modeling Process. Finally, we discuss Perspectives in the context of Object Oriented Modeling (OO) (ref), the Unified Modeling Language (UML) (ref) and the Business Process Modeling Notation (BPMN) (ref). Examples and illustrations are created using Perspectives4EA, an MDG Technology Add-Inn for Sparx System’s case tool Enterprise Architect (ref).

Motivation

Different Perspectives

Information Technology has many roles that use many different methods with even more languages and diagramming techniques (ref). This makes it difficult to provide a coherent picture of the system that is being developed. Different roles have their own perspective based on their responsibilities and focus. It is important that these perspectives can be related to and discussed about to prevent differences in expectations and enable compromises between requirements. To facilitate this, Perspectives offers a modeling approach that provides perspectives for all stakeholders in an early stage of the development or change process.

Simple and Understandable

The Unified Modeling Language (UML) is extensive and complex. Moreover, it uses terminology that is only known in the Information System’s domain. This makes it difficult for Business Analysts to participate in further stages of their realization. Modeling in Perspectives is simple. The Modeling Language contains only five different types of concepts,

two types of Diagrams and one type of relation. All these have names that are understood by Business Analysts as well as Information Analysts. Despite Perspective's simplicity, it replaces the UML Class Diagram, Use Case Diagram, Activity Diagram, State Diagram and Component Diagram.

Modeling Agility

With Perspectives the Analyst creates multi-perspective prototypes of Information Systems. To facilitate an iterative approach, the Perspectives Tooling is optimized for modeling speed. With only the minimum of analyst input, a maximum of modeling artifacts is generated. Moreover, from the modeling artifacts, prototypes of the user interface are generated. This enables the validation of the prototypes during a modeling session with the Product Owner or other Business Representatives. First experiences indicate that this vastly improves the efficiency of the modeling process.

The Role of Contexts

Context is the stage on which events occur and actions are performed (ref.). Roles, events and Actions are the foreground of which Context is the background. The context helps to understand them by putting them in perspective. A clear example is disambiguation of word-meaning.
Another role of Context is scoping. An example is variable scoping in programming languages. Private properties are only defined inside functions or objects, public properties are visible outside of these.
Contexts can be embedded in other contexts. Embedded Contexts are more specific than their Contexts. More specific Context indicate more specific Properties of Roles and more specific Actions of Agents.

Dynamic/ Adaptive Case Management

Although deterministic processes can be modeled in Perspectives, it is particularly suited for the development of Dynamic and Adaptive Case Management Systems. Dynamic in that the Actions, the user can perform are dependent on the Context and adaptive in that the Actions are also dependent on the role the User has in a Context.

The Perspectives Modeling Language (PDL)

The Perspectives Modeling Language is a graphical modeling language. It includes five basic types of artifacts:

- Contexts,
- Roles,
- Actions,
- Views and
- Properties

Contexts

Contexts have Roles and Roles have Properties. Contexts do not have Properties themselves. The Public Properties of a Context are defined in the External Role of a Context, the Private Properties of a Context are defined in the Internal Role of a Context.

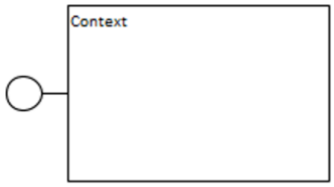

Figure XX: Context with its External Role

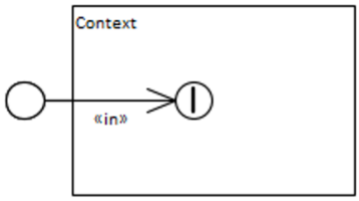

Figure XX: Context with its External and Internal Role

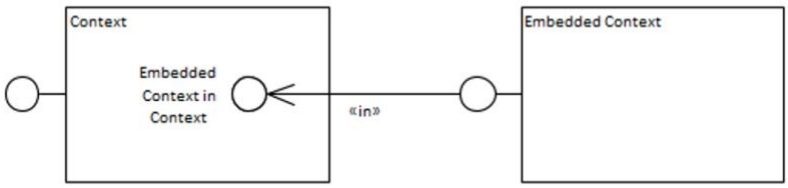

Figure XX: Right Context Embedded in Left Context

Roles

Roles are the “Entities” of the PDL. Roles contain the Properties that represent their state. There are two types of Roles. Ordinary Roles and Agents. Agents are User Roles or Bots. In contrast to ordinary Roles, Agents can be the Subjects of Actions. User Roles and Bots can perform Actions in a Context, ordinary Roles cannot.

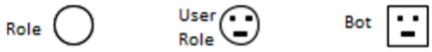

Figure XX: Role, User Role and Bot

The “In” Relation

There is only one type of relation in a Perspectives model; the “In” relation. This asymmetric relation represents the embedding relation between Roles. In Perspectives, we say that a

Role is “filled” by an embedded Role. The “In” relation is represented with an arrow from Role to embedded Role.

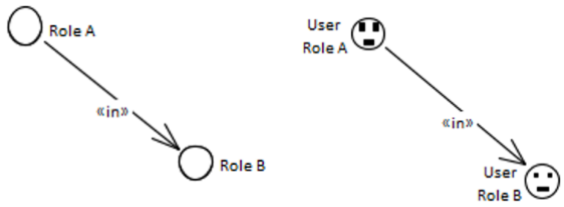

Figure XX: Roles fill Roles

Views and Properties

Roles have Views and Properties. A View is a selection of Properties of a Role that is relevant in the context of specific Actions[1]. An example is the Contact Information of a Person Although a Person may have many properties, only the Properties comprising the Contact Information are relevant when communicating with them.

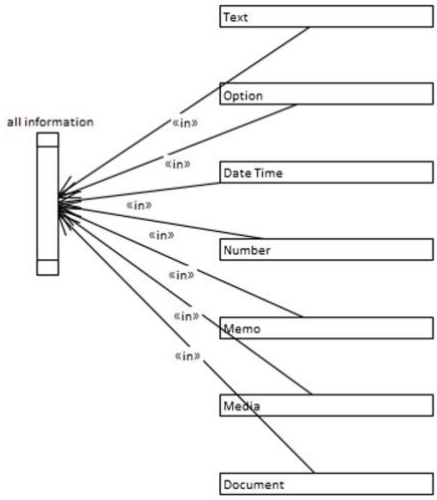

Figure XX: View with Properties

When a Role is filled by another Role, the Properties of the first will be available in the second. This corresponds but is not the same as the notion of “inheritance” in Object Oriented Modelling.

[1] The Perspectives View is analogous to the View in Database Management Systems (DBMS) (ref)

Actions

Actions constitute the behavioral aspect of the PDL and are defined in a graphical form as well as linguistic terms. Actions have a Subject Role and an Object Role. The Subject Role can be filled with Agents (User Roles or Bots), the Object Role can be filled with all three types of Roles that is Roles, User Roles and Bots. Note that when in a Context a User Role or Bot will only fill Object Roles of Actions, they cannot perform Actions themselves in that Context but are only the Objects of other Agent's Actions.[2]

Figure XX: An Action with a Subject Role and An Object Role

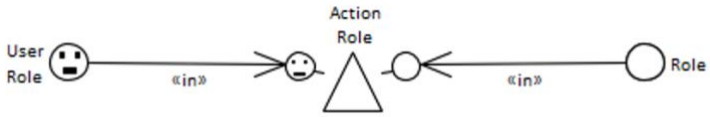

Figure XX: Action with a filled Subject Role and a filled Object Role

In the PDL, all Roles that fill Subject- and Object Roles of an Action must be present in the Context of the Action. When an Object or Subject of an Action is outside the Context of the Action, which is often the case, an additional Role must be specified in the Context of the Action that is filled with the appropriate Roles outside the Context.

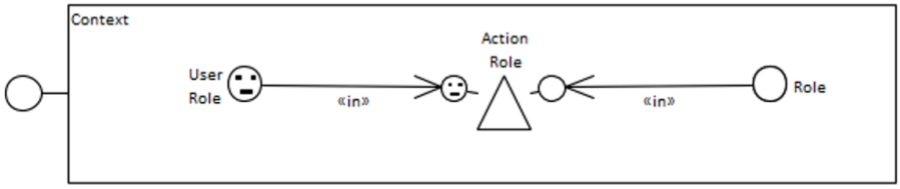

Figure XX: Action with filled Subject- and Object Roles in a Context

In the PDL all Roles that fill Subject- and Object Roles of an Action must be present in the Context of the Action. When an Object or Subject of an Action is outside the Context of the Action, which is often the case, an additional Role must be specified in the Context of the Action that is filled with the appropriate Roles outside the Context.

[2] Action may also have indirect Objects. Then a third Action Role is added that can be filled with any type of Role, dependent on the semantics of the Action.[2] Actions with Indirect Objects are not in scope of this paper and will be discussed elsewhere.

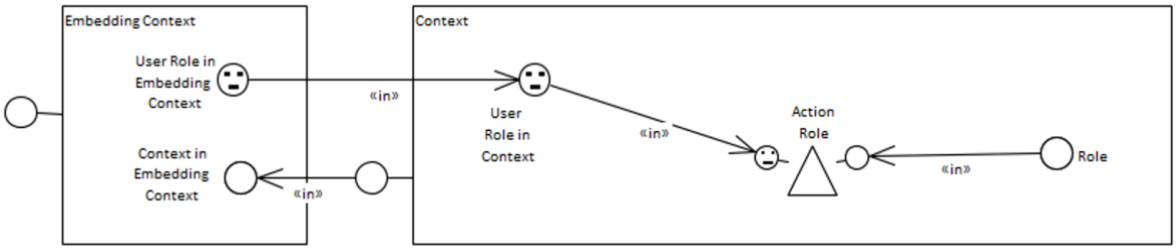

Figure XX: Action with Subject Role filled by local Role that is filled with Role from outside the Action's Context

Calculated Roles and Properties

Roles as well as Properties, can be calculated that is, be derived from other Roles and Properties. Calculating the value of a Property can be the result of the execution of a query, composed of Perspectives Functions, or a foreign function that is performed outside the context of Perspectives. A calculated Role is usually a selection query executed on a Role from one of the embedding Context.

Summary of the Perspectives Modeling Language

The PERSPECT Modeling Language can be summarized as follows:

1. Contexts have Roles and Roles have Properties
2. Contexts have an Internal Role for Private Properties and an External Role for Public Properties
3. Roles are filled with other Roles that together form the Role Chain of a Role
4. A Role contains all Properties that are defined in the Role itself or Properties that are defined in all Roles in its Role Chain
5. Actions have fixed Roles: Subject, Object and possibly Indirect Object
6. The Subject Role of an Action is filled by one or more Agents, the Object Role of an Action can be filled by only one Role of any type: Role, User Role or Bot
7. A View of a Role is a selection of the Properties of that Role, that is relevant in the Context of an Action
8. The View of an Action defines the minimal set of Properties of a Role that is required for the Role to be the Object of the Action
9. A Role can only be the Object of an Action when the Roles contain the Properties that are part of the view of an Action
10. Roles and Properties can be calculated based on other (calculated) Roles and Properties

Types of Contexts, Roles, Actions, Properties and Views

Although the number of different artifacts in the PDL is quite limited, it includes a larger number of different types of these artifacts.

Contexts

Perspectives provides seven types of Contexts:

- **Activities:** execution contexts that have a time and place and a Role that performs the Activity
- **Cases:** management contexts in which the Activities are managed. Typical Examples are a Process or a Project
- **States:** contexts within Cases and Activities that represent the status of a Case or Activity. States determine which Actions Actor-roles can perform given the status of a Case or Activity
- **Parties:** groups, teams or organizations that provide the Actors that perform Actions in Cases and Activities
- **Domains:** container contexts that contain Parties and Cases that belong together in a domain. Examples are the Healthcare Domain or the Personal Domain
- **Aspects** are reusable collections of Perspectives that can be imported in a Perspectives project. In contrast to the Roles and Actions of a Context, Aspect artifacts will not be instantiated

Roles

Perspectives provides three types of Roles:

- **Roles:** any Role that can serve as the Object of an Action. Only Roles contain Properties. The Properties of a Context are defined in the External (public) and Internal (private) Role of a Context.
- **User Roles:** represent the Role the User will have in a Perspective model. User Roles can only be filled by other User Roles
- **Bots:** represent the System. Per definition the Actions of a Bot are automated and will be performed as soon as their Condition is fulfilled. Together with the User Roles, the Bots represent the Agents in a Perspectives model. Only Agents can be the Subjects of Actions. Obviously, Agents can also be the Objects of Actions.

Actions

Perspectives provides seven types of Actions:

- **Consults:** consultation of the Property values of the View of the Role that represents the Object of the Action
- **Changes:** changing of the Property values of the View of the Role that represents the Object of the Action
- **Creates:** a new instance of a Role or a Context. The associated Action View represents the Properties that must be provided during the creation
- **Removes:** removes an instance of a Role or Context. The instance can be added to a different Context if appropriate. Together, these four Actions represent the Information Actions in Perspectives. They are the Perspectives's version of the well-known Create, Read, Update and Delete (CRUD) actions of Information Systems

- **Action** is the generic Business Action in Perspectives. The analyst creates these Business Actions during the modeling process, provides their names and eventually decomposes the Business Actions into the necessary Information Actions associated with the Business Action.

Properties

Perspectives provides seven types of Properties:

- **Text Property**
- **Option Property**
- **Number Property**
- **Date-Time Property**
- **Memo Property**
- **Media Property**
- **Document Property**

Aspects

In PDL, Aspects are collections of Perspectives that represent a certain generic aspect in a Context. Aspects extend the capabilities (Actions) of the Agents (Users or Bots) in a Context. An example is the Location Aspect. This Aspect contains a LocationUser that has Actions that pertain to geographical options such as consulting a location on a map. By filling a User Role with LocationUser the User Role's perspective is extended with the Actions of the LocationUser.

In general, Aspects represent functionality that is orthogonal to the functionality primary to the model's context. In contrast to Roles and Actions of a Context, Aspect artifacts will not be instantiated and can therefore be called "Abstract Contexts". This in analogue to Abstract Classes in Object Oriented Modeling.

Diagrams

The PDL includes two different Diagrams, the Context Diagram and the Role Diagram. The Context Diagram contains the Roles, Actions and embedded Contexts of a Context, The Rol Diagram contains the Views and Properties of a Role.

Conclusions

With only one Connector type, one View type, three Role types, five Action types, six Context types and seven Property types, the Perspectives Diagram Language defines a limited set of diagram symbols that are the constituents of Perspectives models.