Towards Coping with *Role-Binding Anomaly*

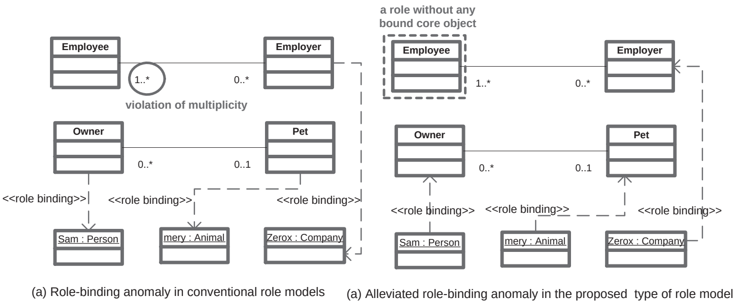
Joon-Sang Lee[1], Doo-Hwan Bae[1], and Ikju Han[2]

[1] Department of Electrical Engineering and Computer Science
Korea Advanced Institute of Science and Technology
{joon, bae}@se.kaist.ac.kr
[2] Department of Computer Engineering Korea Polytechnic University
ijhan@kpu.ac.kr

1 Role-Binding Anomaly

Every role is designed for a specific collaboration, so it can work rightly in the expected context. For example, according to the two role models presented in (a) of Fig. 1, the role *Employer* is modeled to associate with the role *Employee* by *one-or-more* multiplicity, so all of its instances must not be permitted to be handled without an associated instance of the role *Employee*.

Unfortunately, the previous work do not address this issue in depth, so always they have the possibility of allocating roles partially or incorrectly, and missing some preparatory processing before working the roles. Thus, they can allow to manipulate a role or role model in violation of its integrity. We identify this problematic property of roles as *role-binding anomaly*. In this sense, *role-binding anomaly* can be defined as “An problematic phenomenon results in that the structural and behavioral constraints defined in a role model can be violated due to the carelessness of users during role-binding phase, applicable to any mechanism of designing and implementing roles or role models.”.



(a) Role-binding anomaly in conventional role models

(a) Alleviated role-binding anomaly in the proposed type of role model

Fig. 1. Coping with role-binding anomaly.

A. Yonezawa and S. Matsuoka (Eds.): REFLECTION 2001, LNCS 2192, pp. 272–273, 2001.

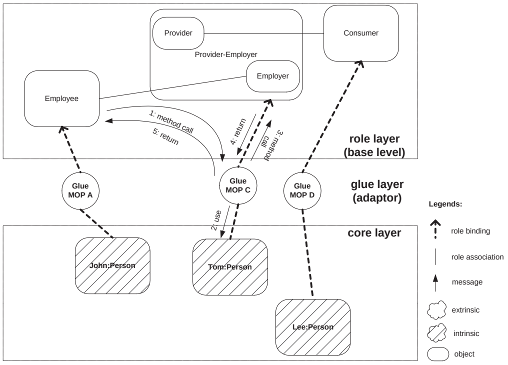


Fig. 2. The enhanced type of role model: overall structure and protocol.

2 Our Proposal

To address this anomaly, we propose an enhanced type of role model, where the system consists of three layers: role layer, glue layer, and core layer as shown in Fig 2. Each layer provides an independent development environment. All classes in the core layer and role layer can be developed without any strong dependency on other layers, and at last glue classes are developed based on MOP, depending on classes in other two layers. The dependency in conventional roles migrates to the glue classes largely. Therefore, the proposed type of role model supports *role encapsulation* in the sense that a core class model and a role model can be developed and tested separately, and also they allow a late binding of each other, acting as software architecture in component-based software developments. We've developed Java *role* package for the proposed type of role model in Javassist 0.8.

Note that the additional properties of roles (*Isolation of core classes*, *Independence of roles*, *Role encapsulation*, *Separation of concerns between Inter-protocol and Intra-protocol*, *Binding of core objects to role instances*, and *Roles as the first-class entity*) defined in our work help alleviate *role-binding anomaly* more than the conventional role model. *Inter-protocol* describes the interaction way among roles, on the other hand, *Intra-protocol* plays a role of the adaptor between role objects and core objects as a glue object. For illustration, an enhanced role model depicted in (b) of Fig. 1 gives a role-binding mechanism to alleviate *role-binding anomaly* largely, providing *role encapsulation.* Even in case that every roles in a role model has no bound core object, the collaborative protocols defined by the role model can be realized by themselves. This benefit results from dealing with roles according to the inherent properties of role and core objects that the core object describes only intra-object behaviors, on the other hand, the role object inter-object behaviors. Allowing roles to form a basic system network, rather than core objects to do so, helps preserve the integrity of system better.