Research on IFML Modeling Considering Session Data

Buye Lou*
School of Management and Engineering, Capital University of Economics and business, Beijing, China

Abstract. The IFML standard does not include the concept of session data, which makes it sometimes difficult for the IFML model to accurately describe the source, destination, formation, and sharing of data for various elements of the web application front end. By extending the IFML metamodel, this paper introduces three new concepts and elements: Session Data, Using Session Data Flow and Providing Session Data Flow. The Using Session Data Flow points from the session data element to the interaction flow element, indicating that the interaction flow element uses the session data. The Providing Session Data Flow points from the interaction flow element to the session data element, indicating that the interaction flow element initializes or updates the Session data. Case studies show that the introduction of these three new elements in IFML model will make the representation of other interaction flow elements more intuitive, accurate and logical.

1 Introduction

In view of the difference between Web applications and traditional application software[1], many development methods for Web applications have been proposed, including HDM, RMM, OOHDM, WebML, WAE, and UWE, etc. most of these methods focus on the modeling of page content and navigation between pages.

Compared with the early websites, the navigation of web applications should reflect the dynamic of web pages and the functionality of web applications. Many web application development methods focus on navigation modeling and design, and put forward some important ideas. For example, OOHDM[2-3] proposed the concept of navigation context, which can provide context constraints for the target objects of navigation. WebML[4] builds navigation model based on content unit, and navigation chain is divided into contextual links and noncontextual links, in which contextual links represents navigation that needs to carry some information. WAE[5] expresses the dynamics of Web pages and the functionality of Web applications through the stereotyped class <<Server Page>>, <<Client Page>> and <<HTML Form>>. UWE[6-7] introduces the stereotyped class <<process class>> and <<process link>> on the basis of the stereotyped class <<navigation class>> and stereotyped association <<navigation link>>, so that Better express the data processing functions involved in the navigation process.

In 2013, the Interaction Flow Modeling Language (IFML), which evolved on the basis of WebML, was proposed[8-9] and was adopted by OMG as the standard modeling
language for the front-end of web applications. Compared with most previous modeling methods, IFML maintains a relative balance between the dynamics, navigation, and functionality of Web applications. Instead of emphasizing the modeling of a certain aspect (such as navigation), IFML organically combines the three aspects. In recent years, many scholars have done a lot of research on the application and practice of IFML. For example, what role does IFML play in the software life cycle, and how to develop a software system in collaboration with other modeling languages [10]; how to extend IFML to adapt to the development of RIA[11], or to adapt to the development of mobile applications[12-13], etc.

Session data is temporary data generated when a web application is running. It occupies an important position in a web application and often directly affects the presentation of the front-end content of the application, the processing of business logic, and the navigation. The IFML standard does not support the concept of session data, resulting in the model cannot reflect the actual situation. This article proposes a method to extend IFML to support session data and introduces its modeling application.

2 Core features of IFML

IFML supports the platform independent description of graphical user interfaces for applications accessed or deployed on such systems as desktop computers, laptop computers, PDAs, mobile phones, and tablets. The focus of the description is on the structure and behavior of the application as perceived by the end user. The description of the structure and behavior of the business and data components of the application is limited to those aspects that have a direct influence on the user’s experience [9].

An IFML model contains an interaction flow model, a domain model, and may optionally contain viewpoints. The interaction flow model is the user view of the entire application and consists of interaction flow model elements. A domain model represents a view of an application's business domain and is a description of the relevant data and behavior in the business domain that can be accessed or referenced in the interaction flow model. A viewpoint represents a particular aspect of the system by referring to a collection of elements of the interaction flow model.

Using IFML to model the user interface and interaction can mainly address the following problems.

- View structure: consists of the definition of view containers, their nesting relationships, their visibility, and their reachability.
- View content: consists of the definition of view components, i.e., content and data entry elements contained within view containers.
- Event: Support user interaction and corresponding event commands.
- Action reference: A reference to an action triggered by a user event. The action represents a piece of business logic, which can be located on the server side or the client side.
- Navigation: The impact of user interaction and action execution on the state of the user interface.
- Parameter binding: consists of the definition of the input-output dependencies between view components and between view components and actions.

IFML can be extended using UML’s extension mechanisms, such as defining stereotype, token values, and constraints [14]. The IFML metamodel is divided into three packages: core package, extension package and data type package. The concepts in the extension package are formed by extending the concepts in the core package. For example, Menu and Window are stereotyped ViewContainer. List, Details, and Form are stereotyped ViewComponent; Select, Submit are stereotyped view element Event, etc.
3 Extend the IFML metamodel for session data

3.1 Session data and its scope

Here, session data refers to temporary data that is stored in memory and generated during a user's session with a Web application, not just temporary data with session scope. In web applications, in addition to persistent data stored in the database, session data is also essential, and it is an important part of the runtime state of the web application. Session data will have a direct impact on the dynamics, navigation, and functionality of web applications.

- **Dynamics.** The content to be rendered on the page comes from session data, which is generated dynamically.
- **Navigation.** Session data affects the flow of navigation; for example, when session data about the logon user information does not exist, the application will navigate to the logon page.
- **Functionality.** Web applications cannot do without business data processing, which often produces intermediate results that may need to be shared across a series of requests. Session data is the best way to exist for these intermediate results.

Session data generally has a certain scope. Different dynamic web technologies use their own mechanisms to manage session data. For example, in JSF, managed bean tools are used to manage session data, with application scope, session scope, view scope, request scope, etc. [15]. In PHP, the management of session data is implemented based on the session mechanism and registering session variables. The scope of these session data is the entire session period, but all these session data can be explicitly destroyed by calling a function. However, from the perspective of application requirements, the scope of session data is not necessarily consistent with the scope currently supported by various technologies. In other words, the scope of various technical support does not necessarily meet the actual requirements of the application background. For example, some session data only needs to be valid during the execution of a use case, which is generally narrower than the session scope and wider than the view or request scope.

3.2 Extend the IFML metamodel

The IFML standard does not include the concept of session data, nor does it define related elements. The data involved in the IFML interaction flow model mainly comes from two aspects: one is the class or entity type defined by the domain model, which is mainly regarded as persistent data; the other is the parameters of the interaction flow element, which can be passed between the interaction flow elements. Due to the lack of the concept of session data, the IFML interaction flow model sometimes has difficulty in accurately describing where data comes from, where it goes, how it is formed, and how it is shared. For example, the data to be rendered by a view component is session data that has already been generated, rather than the data obtained by filtering the persistent data through parameters; An action element updates session data or persists session data, etc.

Next, the concept of session data and related elements are introduced to the IFML interaction flow model by extending the IFML metamodel, as shown in Figure 1. Among them, the metaclass without background color is already defined in the core package of the IFML metamodel, and the three metaclasses with light gray background are newly defined, namely session data, using session data flow and providing session data flow. Here, the resulting interactive flow modeling language is called
SD-IFML. Like view elements, actions, events, parameters, and so on, session data is an IFML interactive flow model element. Session data can be saved on the server side or on the client side. Session data is a multiplicity type element, and its instances contain values. Session data can be a basic type or a complex type, such as an object or a collection of objects.

A session data is always associated with a view container, and a view container can be associated with multiple sessions data. The view container is the scope of the session data associated with it, which can be accessed by each interaction flow element within the view container, including actions caused by events. In IFML, the view container is used to describe the view structure. From a small perspective, the view container can describe the composition structure of a page and a window. From a large perspective, the view container can also describe the view composition structure of a system, a subsystem, or a use case[16]. Therefore, defining a view container as the scope of session data can make the scope choice wider, more flexible, and more suitable for the actual needs of the application.

In the SD-IFML interaction flow model diagram, an unsealed box on the right represents a session data element. Among them, the left side specifies the scope of the data, such as the name of the subject, use case, or the name of the view container, and the right side specifies the data name, type, as shown in Figure 2. A using session data flow means that an interaction flow element reads and uses a session data. An interaction flow element can have multiple using session data flows, which means that it needs these session data to support it to perform specific functions. A providing session data flow means that an interaction flow element transmits data to a session data element to initialize or update the session data. An interaction flow element can have multiple providing session data flows, which means that it can create or update multiple session data.
In the IFML interactive flow model diagram, the navigation flow is represented as a solid line with a triangle arrow, and the data flow is represented as a dashed line with a triangle arrow. In contrast, the using session data flow and providing session data flow are represented as dashed lines with plain arrows. The arrow end of using session data flow is interaction flow element, the end is session data element, the arrow end of providing session data flow is session data element, and the end is interactive flow element.

4 Case study

Master's dissertation process management information system is a management software developed by the author with the help of IFML, which can be used in universities or secondary teaching departments, and can provide support for the process management of master's degree thesis. The system mainly provides process management support for thesis opening, mid-term inspection and thesis review. Here, take the use case of "creating an opening activity" as an example to introduce the establishment of its SD-IFML interaction flow model, as well as the use and effects of session data elements in it. The business related to dissertation opening includes: students uploading the opening report, the administrator grouping for the opening defense, students viewing the defense notice and grouping situation, the judges viewing the grouping situation and reviewing the opening report of students in this group, and the defense team giving each student's report defense results and opinions, students and tutors check the defense results, etc. The prerequisite for these business processes is that the administrator must create an opening activity in advance, and the above business processes are all part of a certain opening activity.

The goal of the "creating opening activity" use case is to create a new paper opening activity, set the corresponding basic properties for the opening activity, and specify the students participating in the opening activity. Its SD-IFML interaction flow model is shown in Figure 3, and its process is roughly divided into three steps.

![Figure 3. SD-IFML model for the use case of "Creating opening activity".](image)

First, the system renders the "opening activity attributes" page view, which contains a form. The administrator can set the basic attributes of the opening activity such as title,
subject, grade, kind (one-time opening/second-opening), deadline for submitting the opening report (deadline1), and deadline for submitting the defense results (deadline2) through the form. When the administrator submits the form, it will navigate to the "initialize" action. The function of this action is to: (1) save the basic attributes of the opening activity to the session data element kt. The kt is an instance of type KT. The Kt is a class in the domain model, which represents the opening activity; (2) Initialize the students participating in the activity, that is, obtain relevant student data from the persistent data according to the subject, grade and kind attributes, and storing them in session data elements stus. The stus is a collection of instances of the type Student, and Student is a class in the domain model.

Next, the system presents the "participating students" page view. This page contains the students list view component, which displays all the students in session data stus. At this time, the administrator can adjust the students individually according to the actual situation: (1) selecting a student from the student list will trigger an event and navigate to the "delete" action. This action will delete the specified student from the session data stus, and then render the current page again (ignoring the navigation flow); (2) Trigger the event of adding a student to open a modal window in which the administrator can enter the student id in the form and navigate to the "add" action when confirming. This action adds a designated student to the session data stus and then re-renders the "participating students" page view.

Finally, in the "participating students" page view, you can trigger an event to navigate to the "create" action. This action reads the session data kt and stus and persists them, achieving the goal of creating an opening activity. After the action function is completed, the action completion event will be triggered and navigate to the use case of "selecting opening activity".

There are two session data kt and stus in this model, whose scope is the whole use case. In the model diagram, the two session data elements are represented not only at the top of the use case model diagram, but also at the bottom of the use case model diagram, where the ordinal number refers to the number of repeated representations of the session data elements. The purpose of this repeated representation is to avoid excessively long using session data flow or providing session data flow and make the model diagram more concise.

5 Conclusions

By extending the IFML metamodel, this article introduces three new concepts and elements of session data, using session data flow, and providing session data flow to the IFML model. The scope of session data is the view container associated with it, which is more flexible than the scope supported by general dynamic web technology and can meet the actual requirements of the application. Using session data flow refers to the interaction flow element from the session data element, indicating that the interaction flow element uses the session data. Providing session data flow refers to the session data element by the interaction flow element, indicating that the interaction flow element initializes or updates the session data. With the help of these three new concepts and elements, IFML interaction flow model can more accurately describe the data source, destination, formation and sharing mode of relevant elements in the application front end.
References


413