A Knowledge Abstraction Approach for Multimedia Presentation

Timothy Shih, Louis R. Chao, Chi-Ming Chung, Y.H. Wang, Wei-Chuan Lin, W. C. Pai
Department of Information Engineering
TamKang University
Taipei 25137, Taiwan, R. O. C.
E-mail: g1320039@cs.tku.edu.tw

Abstract
The demonstration of multimedia presentation can be promoted by using multi-vendor's tools. The more tools are used, the more complicated communication is needed among these tools. The integration of these multimedia presentation tools is thus important. This paper describes an architecture named Tool Integration Platform (TIP) to integrate tools in a knowledge abstraction way. TIP is composed of a CID (Control Integration Daemon), a CII (Control Integration Interface) and some Integration Inference Rules (IIR) that are applied by the Integration Inference Engine (IIE). The IIR are stored in a Repository and used to deduce tool knowledge dynamically. In this way, many tools can be integrated into a cooperative multimedia presentation developing environment. To verify this architecture, a number of multimedia tools are integrated into TIP. Finally, an integration assessing method is used to evaluate the integration status of tools in TIP.

Keywords: Tool Integration Platform, Integration Inference Engine, Integration Inference Rule, Repository, Control Integration Daemon

1. Introduction

Multimedia presentation is critical to demonstrate the effect of multimedia. The objective of each multimedia tool is to increase the productivity, provide the better view and simplify the multimedia development. Because users have their own preferred tools developed by different tool vendors, it is important to integrate those heterogeneous tools in a cooperative developing environment. To support multimedia across open distributed systems, all of the tools should have the appropriate Client/Server architecture. However, a company does not have to develop all the tools to meet users' requirements. The tools should cooperate to compensate each other's weak-points. For example, the Resource Editor (RE), Resource Browser (RB) and Presentation Designer (PD) tools are widely used to capture and display the multimedia resources respectively. When developing a multimedia presentation, the planner may use these tools to prepare the presentation resources. Therefore, the RE, RB, and PD tools should be integrated together. Thus, when the resources captured by RE, they are sent as the input data of the RB and PD tools automatically, and demonstrate the multimedia presentation. In this way, the job for developing multimedia presentation would be convenient. The automatic processes can be done in the same way in different multimedia developing steps via many integrated tools.

In this paper, section two describes the approaches for tool integration. Section three explains the major components of the proposed tool integration architecture called TIP. Section four is partitioned into three parts. The first part describes the verification of TIP through integrating a set of multimedia tools. The second part introduces the functions and relation between the IIE and IIR. The third part proposes an integration assessing method is used to evaluate the integration status of tools in TIP. Section five is the conclusion and the continuing research.

2. The Tool Integration Approaches

There are three evolving approaches for tool integration. The first is called brute-force approach which integrates a set of predefined tools and forms a cooperated tool environment. However, the way for exchanging data among tools is used the Import/Export functions without taking the data semantics into consideration. Thus, if there is a new tool which is planned to join to the integrated environment, what will be the relation between the new tool and the pre-integrated tools as drawn in Fig. 1?

The second is called vendor dependent approach which integrates a set of tools that are developed by the same tool vendor and also named as Integrated
CASE (ICASE) tools approach. The advantage of this kind of tool integration is the tools are optimally integrated. However, the semantic data cannot be exchanged among different vendor's tools. Some ICASE vendors are attempted to integrate with other vendor by opening their metamodel of tools. This phenomenon can be shown as Fig. 2. The well-known environments for ICASE tool approach are TI (Taxas Instrument)'s IEF, DEC's FUSE, and IBM’s AD/CYCLE[17].

3. The Platform of Tool Integration

The architecture proposes in this paper is called Tool Integration Platform (TIP) which can be expressed as a set of transformation functions to map a tool to other tools. After the service has been done by tools, the transformation functions can transfer control back to the original tool. The mapping is denoted in a Finite State Machine like manner as:

\[ TIP = (Q, \Sigma, \delta, T, O) \]

where

- **Q**: A finite set of internal states, including \{Active Run (AR), Not Run (NR), Background Run (BR)\}
- **\Sigma**: A set of input such as \{resource ...
- **T**: A set of tools such as \{RE, RB, PD ...
- **O**: The output set of tools such as \{reviewed resource, generated presentation ...
- **\delta**: A set of transition functions include \{provide, listen, send, notify\} and can be denoted as:

\[ \delta: Q \times \Sigma \times T \rightarrow Q \times T \times O \]

For example:

\[ \delta_{\text{new}}(\text{AR}, \text{“resource”}, \text{RE}) \rightarrow \{\{BR, NR\}, \{RB, PD\}, \{\text{reviewed resource, generated presentation}\} \}

This means that the running tool RE sends the “resource” to RB or PD. These two tools are originally in not running or background running state. They are triggered to execute the service of reviewing the resource or generating multimedia presentation. To achieve this transformation, TIP is divided into five components which are Control Integration Daemon (CID), Control Integration Interface (CII), Integration Inference Rules (IIR), Integration Inference Engine (IEE), and the Repository. With these mechanisms, TIP is a machine independent platform which can integrate any kind of multimedia tools. These mechanisms are explained as followed:

- The CID is the message server which dispatches the message to the suitable tools and triggers the IIE to apply the stored IIR suitably. The CID is the interface used to integrate tools into TIP.
- The IIE is the inference engine in TIP for deducing tool knowledge. It is triggered by CID when a tool is registered in TIP or a message is sent to CID. In this way, the new deduced tool knowledge is produced and stored in the Repository.
- The IIR are inference rules applied by IIE and
stored in the Repository. The IIR are the basic inference rules in TIP and can be extended by adding new IIR.

The Repository is used to store the tool knowledge and IIR. The tool knowledge includes: the registered tool name, basic and extended definitions for IIR, and the System Default Configuration File (SDCF) which contains the system tools.

The architecture of TIP is to enhance the standard proposed by ECMA/PCTE to provide an new integration environment[7,9] drawn as Fig. 5.

![Fig. 5 The Proposed TIP Architecture](image)

4. Integrated Tools, Integration Inference Rule, and Assessment for TIP

4.1 Tools integrated in TIP

To verify the feasibility of TIP, many tools such as the Resource Editor(RE), Resource Browser(RB) and Presentation Designer(PD)[15,18] are integrated in this environment. Other system tools such as the tool manager(CIP Manager) and message monitor(Monitor) are also implemented in the TIP to monitor the tool invocation and the flow of message-passing.

The traditional multimedia developing flow is shown in Fig. 6. The developers have to use these tools step by step to develop a multimedia presentation. Thus, these tools should be integrated for reducing developers' efforts. The RE includes many editors such as Text Editor, Animation Editor ... etc. to accept the digitized multimedia resources. The RB is used to review the resources accepted from RE and stored in the Resource DataBase. The PD is used to schedule and synchronize the presentation resources. The integration architecture in the TIP can be drawn as Fig. 7. All of these tools are triggered through the message passing to CID of TIP, then, the CID drives the IIE to apply the IIR[20] and deduces the tool knowledge.

![Fig. 6 Tradition multimedia developing flow](image)

![Fig. 7 Current tools integrated in TIP](image)

4.2 The IIE and IIR

For the sake of explaining the IIR in TIP, some mathematical sets are expressed as:

Let $C$ be the domain of input source $\forall C = \{\text{Resource}\}$, $T$ be the domain of tools $\forall T = \{\text{RE, RB, PD}\}$ and $S$ be the domain of services provided by each tool in $T$. The integrated tools of TIP can be denoted as:

$$TIP(ToolSet) = \sum_{i,j,k} T_i(S_j(C_k)),$$

where $\forall i,j,k, T_i, S_j, C_k \in T, S, C$.

This means that the input set $C_k$ processed by the service set $S_j$. The $S_j$ is provided by the tool $T_i$. To achieve this phenomenon, some definitions, which are stored in the Repository, denoting in predicate logic are:

Definition 1: $P(S, T)$ means that tool $T$ provides the service $S$, $T \in T$ and $S \in S$. For example, the RE provides "Save digital resource" service. Therefore, it can be denoted as $P("\text{Save digital resource"}, \text{RE})$.

Definition 2: $L(S, T_1)$ means tool $T_1$ listens the service $S$ which is provided by other tool $T_1$, $T_1, T_2 \in T$ and $S \in S$. For example, the RB listens the "Save digital resource" service provided by Compiler. Therefore, it can be denoted as $L("\text{Save digital resource"}, \text{RB})$.

Definition 3: $I(\text{Text}, T)$ means that Text is the input to tool $T$, $\text{Text} \in C$ and $T \in T$. This definition is applied by IIE to check the run time relations of tools. For example, the RB is used to review a resource. That is a resource is the input to the RB.
Therefore, it can be denoted as \( I(resouce, RB) \).

**Definition 4:** \( O(T, Text) \) means that \( Text \) is the output of tool \( T \). \( Text \in C \) and \( T \in T \). For example, the \( RE \) is used to edit and save a resource. That is a resource is the output of the \( RE \). Therefore, it can be denoted as \( O(RE, resource) \).

**Definition 5:** \( D(T1, T2) \) means that tool \( T1 \) and \( T2 \) have some dependencies, \( T1, T2 \in T \). That is there are something that are shared between tool \( T1 \) and \( T2 \). If the sharable thing is changed in tool \( T1 \), it may influence the tool \( T2 \). For example, a resource is sharable by a \( RE \) and a \( RB \). After editing by \( RE \), the resource may influence the review of \( RB \). In this way, it can be denoted as \( D(RE, RB) \).

**Definition 6:** \( Tri(T2, T1, S) \) means that tool \( T2 \) can be triggered by tool \( T1 \) after the service \( S \) is completed in \( T1, T1, T2 \in T \). That is if the service \( S \) is provided by tool \( T1 \) and listened by tool \( T2 \), tool \( T2 \) can be triggered by tool \( T1 \). For example, \( P("Save digital resource", RE) \) and \( L("Save digital resource", RB) \) then the \( RB \) can be triggered by the \( RE \). After the service \( "Save digital resource" \), \( RE \) has finished the service "Save digital resource". Therefore, it can be denoted as \( Tri(RE, RB, "Save digital resource") \).

**Definition 7:** \( IT(T1, T2) \) means that tool \( T1 \) and \( T2 \) are well-integrated. \( T1, T2 \in T \). That is tool \( T1 \) and \( T2 \) are integrated tightly if they can be triggered by each other. For example, \( Tri(RE, RE, "Save digital resource") \) and \( Tri(RE, RB, "Resource search & not enough") \) then the \( RE \) and \( RB \) are well integrated. Therefore, it can be denoted as \( IT(RE, RB) \).

From the above definitions, the Integration Inference Rules (IIR) can be summarized as:

**Integration Inference Rule (IIR):**

**Rule 1:** \( \exists T1, T2, S, Text \)

\( O(T1, Text) \land I(Text, T2) \land P(S, T2) \rightarrow D(T1, T2) \)

**Rule 2:** \( \exists T1, T2, S \)

\( P(S, T1) \land L(S, T2) \rightarrow Tri(T2, T1, S) \)

**Rule 3:** \( \exists T1, T2, T3, S1, S2 \)

\( Tri(T3, T2, S1) \land Tri(T2, T1, S2) \rightarrow Tri(T3, T1, S2) \)

**Rule 4:** \( \exists T1, T2, S1, S2 \)

\( D(T1, T2) \land Tri(T2, T1, S1) \land Tri(T1, T2, S2) \rightarrow IT(T1, T2) \)

The above IIR are the basic Inference Rules applied by IIE in run time. The IIR can be extended by using the above Definitions to add new IIR. When a tool is intended to be integrated into \( TIP \), the developer should use the \( CH \) to register the tool in \( TIP \). At that moment, the IIE is triggered by \( CID \) to apply the suitable IIR in the Repository to produce the appropriate tool knowledge.

The tools integrated in the \( TIP \) as described in Fig. 7 may apply IIE to deduce the tool knowledge. These tool knowledge can be drawn as a graph which is called Tool Dependency Graph (TDG) and shown as Fig. 8. In TDG, the solid line shows the direct triggering relation among tools. While the dashed line is produced through the indirect triggering of tools. It is obvious that Fig. 8 is the automatic steps of Fig. 6.

![Fig. 8 The Tool Dependency Graph of TIP](image)

**4.3 The Integration Assessing Method**

To evaluate the integration status of tools, an evaluate mechanism is proposed in this paper. The evaluation mechanism -- Quantity metric \( Qm \) to justify the integration status of tools and can be denoted as:

\[ Qm = \left( \frac{\sum \frac{S_i}{ST_i}}{N} \right) \]

where

- \( S_i \) is the used provided services in tool \( T \),
- \( ST_i \) is the total provided services in tool \( T \),
- \( N \) is the total number of tools which are integrated in a cooperated environment.

For example, the \( Qm \) of the Fig. 8. can be computed as followed:

\( RE \) provides 3 services which were used just one of them.

\( RB \) provides 2 services which were used all of them.

\( PD \) provides 4 services which were used just three of them.

Therefore, the \( Qm \) of Fig. 8 is \( \frac{25}{36} \).

In this way, the integration status can have a quantity metric for evaluation.

**5. Conclusion and Continuing Research**
The architecture (TIP) discussed in this paper is to provide an environment to integrated tools in TIP. In this manner, tools can compensate the drawbacks for each other. This architecture also provides the new idea of the IIE which applies suitable IIR to deduce the tool integration knowledge dynamically. The TIP is a tool integration architecture which can be applied not only limited to integrate multimedia tools but also the other fields such as the CAD, CAE, CASE...etc. With TIP, the time required to develop a new tool or modify an existed tool for tool integration can be shortened.

Thus, for continuing research, the verification of interoperability of TIP will be done. That is to integrate tools distributed among the same or different operating systems such as UNIX or OS/2 operating environment.

References


