VINCA – A Visual and Personalized Business-Level Composition Language for Chaining Web-Based Services

Yanbo Han[1], Hui Geng[1], Houfu Li[1], Jinhua Xiong[1], Gang Li[1], Bernhard Holtkamp[2], Rüdiger Gartmann[2], Roland Wagner[2], and Norbert Weissenberg[2]

[1]Institute of Computing Technology, Chinese Academy of Sciences, 100080, Beijing, China
[2]Fraunhofer Institute for Software and Systems Engineering, 44227, Dortmund, Germany
{yhan,genghui,lhfsday,xjh,gangli}@software.ict.ac.cn

Abstract. The paper presents a service composition language called VINCA, which differs from many existing ones in its emphasis on enabling business users to visually "program" from business view-point their personalized applications on the basis of Web-based services. VINCA embodies an integrated approach to mediating between diverse, rapidly changing user requirements and composites of individual services scattered over the Internet. The approach is targeted at application scenarios that require Web-based services be quickly assembled by non-computer professionals to fulfill certain spontaneous requirements. VINCA is developed within a real-world project for developing a service mediation platform for the Olympic Games Beijing 2008, on which an effective information system providing personalized and one-stop information services to the general public, should be based. In this paper, we introduce the main features and design rationales of VINCA with a scenario, and also discuss its implementation and application.

Keywords: Business-level service composition, Semantic Web services, On-demand and just-in-time application construction

1 Introduction

The Internet and the World-Wide Web opened up a new horizon for us to acquire information and also aroused our appetite for a more flexible and scalable computing infrastructure for building up cross-organizational information systems. In this connection, many efforts have in the past years been made on developing technologies and platforms for integrating autonomous, heterogeneous and Web-based services [Casa01][Sing01][Weis01]. The service-oriented technologies that are currently highlighted by the Web Service architecture are among the prominent outcomes. Web-based services for a wide range of purposes are beginning to

The research work is supported by the National Natural Science Foundation of China under Grant No. 60173018, the Key Scientific and Technological Program for the Tenth Five-Year Plan of China under Grant No. 2001BA904B07 and German Ministry of Research and Education under Grant No. 01AK055

M.E. Orlowska et al. (Eds.): ICSOC 2003, LNCS 2910, pp. 165–177, 2003.

proliferate. Service composition will play an essential role in making full use of these services as well as the Internet infrastructure, and a number of composition languages have been proposed [Andr03][Casa01][Kici01][Picc02]. Most of these languages were developed for software professionals and targeted at developing large-scale E-business applications and the like.

Meanwhile, we can also observe the trend towards user-centric and just-in-time application construction. Instead of constructing monolithic and wholly packed applications for business users, software professionals will be requested to concentrate more on establishing IT infrastructure and enabling environments, and to leave the job of configuring individual applications to business users in order to better cope with the diversity of user requirements and the uncertainty of Web-based resources. To date, there exists a wider spectrum of application scenarios that require Web-based services be quickly assembled and coordinated by non-computer professionals to fulfill certain spontaneous requirements. Examples of such application scenarios include dynamic supply chain management, handling of city emergency, and management of massive public events. In fact, we are undertaking a real-world project that bears exactly the same characteristics. The project is called FLAME2008, which is abbreviated from *A Flexible Semantic Web Service Management Environment for the Olympic Games Beijing 2008* [Holt03]. It is a Sino-German joint effort for developing a service mediation platform for the Olympic Games Beijing 2008, on which an effective information system providing personalised and one-stop information services to the general public, should be based. Adopting the service-oriented paradigm, we designed the service integration platform with a novel integrated approach called CAFISE [Han03] to mediating between diverse, rapidly changing user requirements and composites of individual services scattered over the Internet. A central element of CAFISE is the user-centric, business-level service composition language – VINCA, shortened from *A Visual and Personalized Business-level Composition Language for Chaining Web-based Services*.

The paper is organized as follows: Section 2 gives a brief overview of FLAME2008 and CAFISE to make clear the context of VINCA, and then discusses the business-end programming metaphor underlining the language. Section 3 describes the language and its enabling environment. A simplified application scenario is examined to evaluate the design of VINCA. Section 4 sums up with several concluding remarks.

2 Context and Positioning of VINCA

As mentioned earlier, the development of VINCA is driven by a set of practical requirements. To better explain the design rationales of the language, let us first get an overview of FLAME2008 project and the CAFISE approach. Thereafter, in line with the business-end programming metaphor, we discuss the positioning and design considerations of VINCA.

2.1 FLAME2008

As shown in Fig.2.1, FLAME2008 is to mediate between personalized requirements of different user groups (reporters, spectators, organizers, athletes, etc.) through different channels (stationary information kiosks, mobile devices, etc.), and a large pool of services provided by various parties (organizers, governmental organizations, profit-oriented service providers, etc.). Fig.2.2 goes further to illustrate the conceptual architecture of FLAME2008. Here, individual user requirements are resolved as FLAME2008-based applications, which are specific assemblies of Web-based services and are formed directly by business users in an easy and just-in-time manner. The major components shown in Fig.2.2 are to cover the following key aspects:

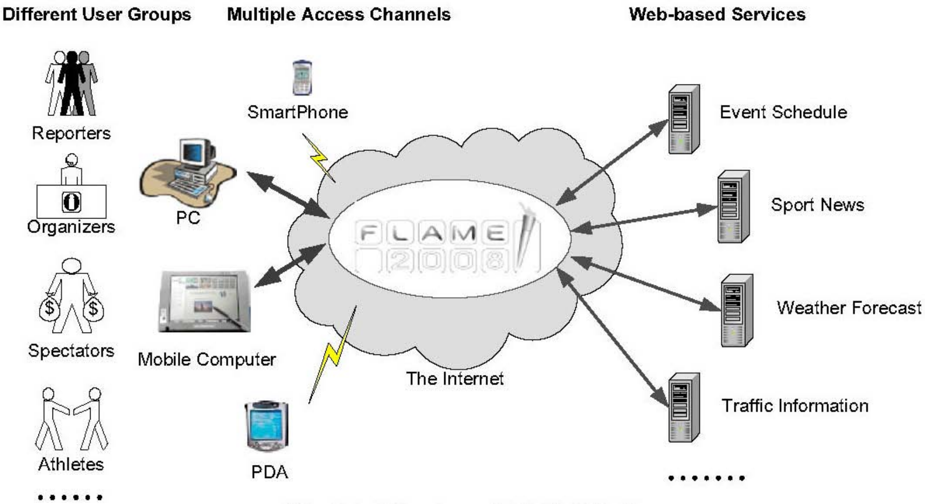


Fig. 2.1. Objectives of FLAME2008

Semantic Service Organization

In FLAME2008, Web-based services are at present based on Web Services. Although Web Service standards provide a basic means to encapsulate and register Web-based services, service registration alone is not enough to get all registered resources well organized for different purposes. FLAME2008 provides a mechanism to enable personalized organization of services on semantic level. The semantic descriptions are used for composing and matching services. As a basis for service semantics description we use ontologies. A more detailed discussion of the approach can be found in [Weis03]. Based on a set of given criteria including user context, services can be logically grouped into service communities with multiple layers. Business-oriented services can then be extracted from service communities.

Business-level Service Composition

For each eligible user, FLAME2008 manages a business-oriented logic view of all Web-based services made available to him or her according to his/her user context. The user can browse the resources and configure them according to their business needs. Through the business-end programming environment, they only need, based on predefined modeling elements and rules, to drag and configure the symbols representing services at different levels of granularity to express their business requirements. Since this issue is the main point of discussion in the paper, a separate section is devoted to it subsequently.

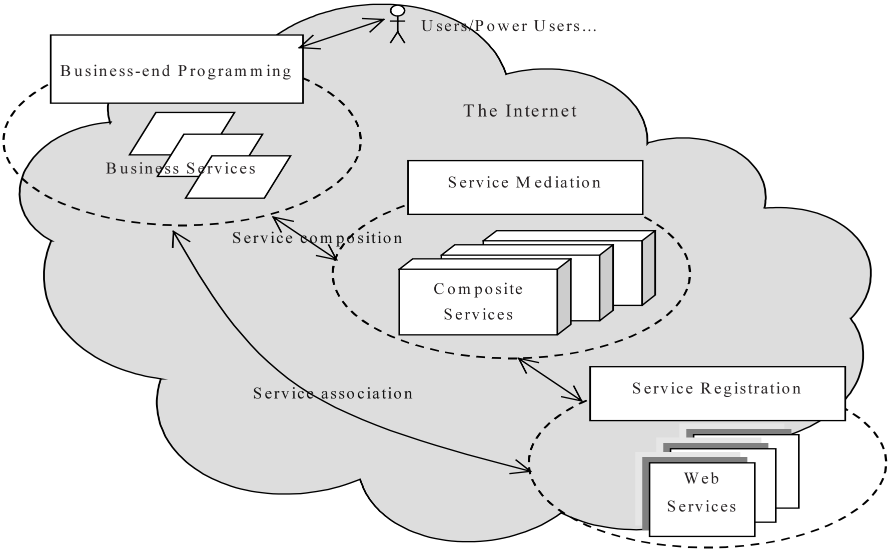


Fig. 2.2. Conceptual Architecture of FLAME2008

Service Mediation

With the help of ontology-based descriptions of service semantics, user demands that are captured with the business-level specifications are either matched directly with service offerings or related to a composition of a number of service offerings. VINCA is the language we developed for the business-level specifications. For the operational level composition of Web services, a subset of the standard Business Process Execution Language for Web Services - BPEL4WS [Andr03] is used. The mapping between our business-level models and BPEL4WS specifications is realized by the FLAME2008 platform. Derived from application scenarios that reflect a typical usage of a targeted user group, composition patterns can be predefined and used as a basis for individual cases. Composition of services can be performed either automatically based on a full-fledged composition pattern and related ontological descriptions or can be defined manually with the business-end programming environment.

2.2 The Business-End Programming Metaphor

The business-end programming metaphor advocates that business-level service composition should not be considered in computer programming logic as we usually do. We call the process, with which a business user configures his or her applications, business-end programming. To stress user involvement, we coined the term – power user. A power user is any business user who knows little about software programming but is capable of configuring his businesses with tool support. FLAME2008 aims at supporting power users in (re)configuring their service-based applications on their demand. Instead of directly introducing object-oriented concepts and technologies into a business domain as in [Sing01], our approach proposes to build an integrated model to cover business issues and software issues at the same time.

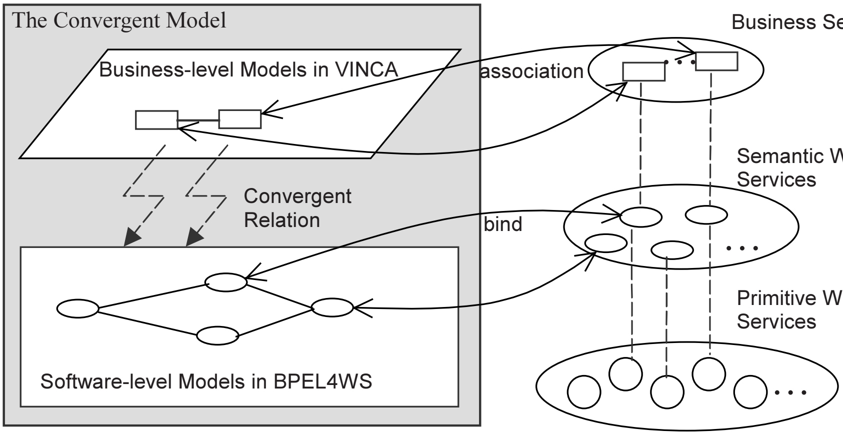


Fig. 2.3. The CAFISE Convergent Model

Highlight of the CAFISE approach is a convergent model - the CAFISE model, which is a grand model covering business issues and software issues simultaneously [Han03]. As shown in Fig.2.3, a CAFISE model has three related parts: a business-level model, a software-level model and a convergent relation as glue. Expressed by VINCA, business-level models are designed to reflect business requirements with a minimum set of business-end programming concepts and mechanisms, which are easy for a business user to understand and master. A business user can (re)configure his/her applications by building or editing business-level models. Software-level models deal with compositions of Web Services. CAFISE narrows the gap between the two levels with a convergent relation. The convergent relation helps to relate or map simple and intuitive business-level elements into more concrete and executable elements at software level.

3 Design, Implementation, and Application of VINCA

3.1 A Simplified Application Scenario

To better explain how VINCA works, the following imaginary example is used for subsequent discussions: In 2008, various parties will provide a large variety of information services for public uses, and VINCA is to help different groups of users to define their personalized “applications”. Among the users is Mr. John Bull, a sport journalist. He is going to write a series of reports for his sport magazine during the Olympic Games. In his spare time, he also plans to visit some famous sights and meet some acquainted peers coming from other countries. Before he leaves for Beijing, he can use the FLAME2008-based information system to schedule his activities and enjoy the multitude of services. For a simplified example, in addition to making some online reservations, he orders his personalized weather information service and can arrange his meetings and sightseeing activities according to actual weather conditions. He can also subscribe information services provided by the organizing committee to periodically get the latest schedule information of press conferences.

3.2 Language Design

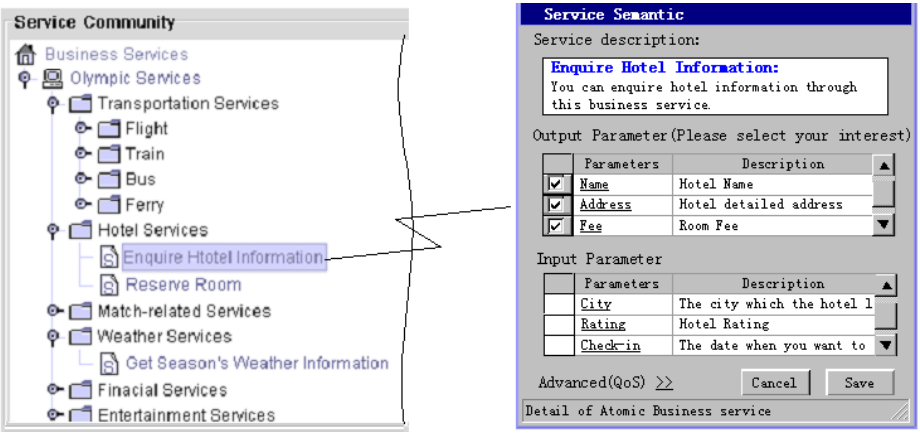
Like some other efforts [Krit01][Shen02], VINCA is a process-oriented language designed for visually capturing user’s needs at a high abstraction level. Based on BPEL4WS, VINCA intends to take a step further towards:

- Adding semantic descriptions to Web services so that they can be composed at business level more easily.
- Organizing semantic Web services rationally into business services so that a business user can easily understand and identify the right ones she needs.
- Letting business users to visually configure business services with easy-to-use operations like drag-and-drop and letting the system to take over related tedious work of interconnecting Web services.
- Defining and making use of user context information so that users get more accurate and personalized information provided.
- Defining interaction channels and user interfaces flexibly.

In the context of the FLAME2008 project, the first working version of VINCA reported in the paper is proposed and implemented, answering partly the above-stated questions. VINCA deals with the following four aspects: service conglomeration, process specification, context specification and interaction specification. Surrounding VINCA process specifications, the other three components provide necessary support. In the subsequent parts, we informally describe these components.

3.2.1 Service Conglomeration

In FLAME2008, different users may “see” different sub-sets of all available resources in terms of her user context including preferences and privileges. As illustrated on the right side of Fig.2.3 shown above, FLAME2008 has three abstraction layers and provides a mechanism called service community [Cafi02] for organizing Web-based services. Each service community corresponds to an organized set of services with domain-specific semantics. Through this mechanism we get a uniform view of Web-based services and their semantics. Semantic services are attributed by a list of semantic descriptions, each of which refers to an ontology concept shared by all involved parties [Weis03]. In a business domain, services are often interrelated with each other in some common patterns. In a service community, a set of supplementary rules can be defined for member services to capture common patterns of interrelationship to improve efficiency or ensure correctness.



Service description:

Enquire Hotel Information:
You can enquire hotel information through this business service.

Output Parameter(Please select your interest)

	Parameters	Description
☑	Name	Hotel Name
☑	Address	Hotel detailed address
☑	Fee	Room Fee

Input Parameter

	Parameters	Description
□	City	The city which the hotel l
□	Rating	Hotel Rating
□	Check-in	The date when you want to

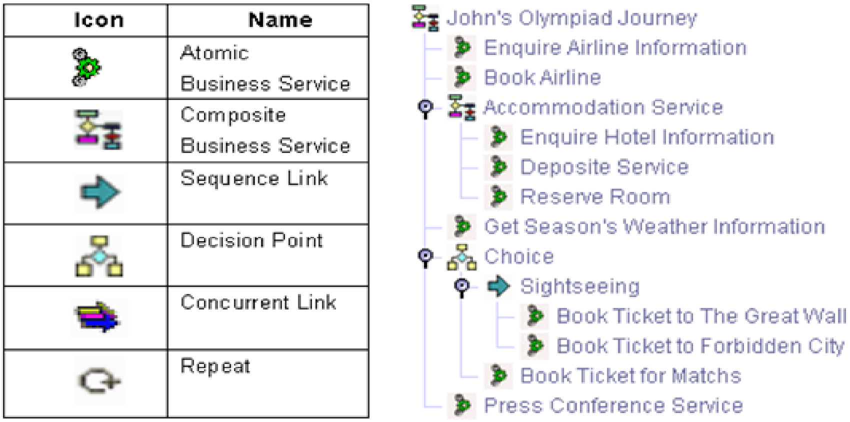
Fig. 3.1. Visualization of Business Services

As Fig.3.1 shows, business services are organized in terms of an application-specific classification for the purpose of navigation. In addition to its semantic description, a business service visually exposes to users a list of input and output parameters that are all in character form with separate type information. When composing services, users can either match the output and input parameters by drag-and-drop corresponding icons or specify certain parameter as “runtime decision”. In the second case, the system will interact with users to acquire information at runtime. Each business service has a separate panel for specifying service qualities. This functionality is designed for more advanced users to specify her QoS requirements, such as cost, time of response and so on. Besides normal business services, there are composite business services that are pre-defined processes and can be used as if it were a normal business service. FLAME2008 cares about its updates in case its constituent member changes.

3.2.2 Process Specification

A VINCA process specification represents business logic from user’s viewpoints. The left part of Fig.3.2 shows the graphical elements of VINCA process. For defining business-level processes, we adopt a Windows-Explorer-alike style. A node can be a service node or a control node, presenting a business service (either atomic or composite) or a control element expressing some control logic. Service nodes in the same column are ordered from top to bottom for activation. Composite service nodes can be unfolded into columns at the next layer, arranged from the left to the right. In order to specify the complete business logic, VINCA employs four kinds of control nodes linking business services: sequential link, concurrent link, decision point and repeat. Service nodes following a sequence link are ordered sequentially from top to bottom for activation. Sequential link is the default linkage mode between services. Service nodes following a concurrent link can be invoked in parallel. Decision point is associated with a condition expression and determines which one(s) of the following nodes can be activated. The control node - repeat – is also associated with a condition and the service nodes following it will be invoked repeatedly until the associated condition is satisfied.

The right part of Fig.3.2 shows partially the process specification of John's application: John's Olympic Journey. "Accommodation Service" is a composite business service, which consists of three atomic business services at the next layer. According to the actual weather status acquired by the service - Get Season's Weather Information, John will decide to watch matches indoors or go sightseeing. The resulting process specification itself is also treated as a composite business service.



Icon	Name
⚙	Atomic Business Service
☰	Composite Business Service
➡	Sequence Link
⋔	Decision Point
⇉	Concurrent Link
⟳	Repeat

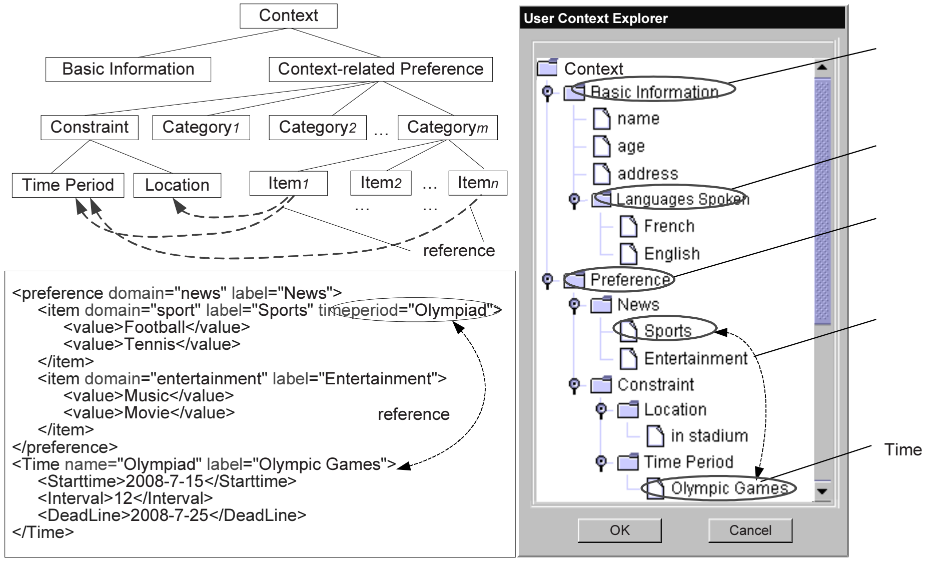
- John's Olympiad Journey
  - Enquire Airline Information
  - Book Airline
  - Accommodation Service
    - Enquire Hotel Information
    - Deposite Service
    - Reserve Room
  - Get Season's Weather Information
  - Choice
    - Sightseeing
      - Book Ticket to The Great Wall
      - Book Ticket to Forbidden City
    - Book Ticket for Matchs
  - Press Conference Service

Fig. 3.2. A Snippet of VINCA Process Specification

3.2.3 Context Specification

One of the most important objectives of VINCA is to move as much as possible the complexity of a business-level specification to FLAME2008. Since a business user pays more attention to what a service does in their business terms, rather than how certain functionality is achieved or which service does the job, we need to hide all the details on the one side, and to supplement enough information for the step-wise refinement and interpretation on the other side. Thus, we have to define an effective context for the business-end programming environment. In fact, context-aware applications are gaining momentum due to the flexibility and extra functionality that context information brings [Brow97][Abow00]. User context is integrated into VINCA and serves as a reference in forming service communities and as a sort of implicit input in defining and executing a VINCA process specification. One VINCA process specification may associate with many context specifications, each of which is associated with a user.

In VINCA, user context is represented and organized in hierarchical, tree-like structure. In the upper-right corner of Fig.3.3, an exemplary user context of John is given: his information is organized in several layers. Internally, user context is expressed in XML. A user context consists of basic information, user preferences, and constraints such as time and location. Users may have multiple preference categories at the same time. Each preference category represents a category of interests. The categories share the same semantic infrastructure of VINCA service communities. The domain attribute can be linked to business service, and the keyword and value in item can be used to specify the non-functional requirements for the business service. Through these attributes (domain, keyword, value), the semantic of basic information or user preferences can be specified, similar to the semantic mechanism of service community. A user context may dynamically include new categories of preference when user interests change.



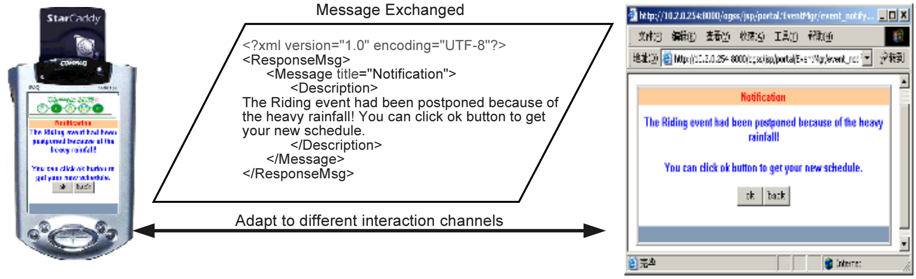
```
<preference domain="news" label="News">
    <item domain="sport" label="Sports" timeperiod="Olympiad">
        <value>Football</value>
        <value>Tennis</value>
    </item>
    <item domain="entertainment" label="Entertainment">
        <value>Music</value>
        <value>Movie</value>
    </item>
</preference>
<Time name="Olympiad" label="Olympic Games">
    <Starttime>2008-7-15</Starttime>
    <Interval>12</Interval>
    <DeadLine>2008-7-25</DeadLine>
</Time>
```

Fig. 3.3. Illustration of a User Context

3.2.4 Interaction Specification

Today, users have multiple choices to interact with an information system. In our scenario, John may interact with FLAME2008 using different devices such as ordinary desktop explorer, PDA, and smart phones in her preferred Also, interaction interfaces for individual services need to be defined. In VINCA, we introduce Interaction specification to deal with the above-mentioned problems. An Interaction specification describes the way of service delivery, and one may also define templates for user interactions.

In VINCA, Interaction specification consists three orthogonal parts: Channel, Template and Mode. Channel defines the terminal or client via which users interact with business application. A channel may be a kind of device, such as Personal Computer, Smart Phone or PDA, or a kind of client software, for instance Email client Outlook. Template specifies formats for capturing or delivering information. Mode describes how users will get or send information, it may be in a pull mode or a push mode. At present, FLAME2008 supports three kinds of interaction channels (PC Internet Browser, PDA Browser and Smart Phone Browser). Fig.3.4 gives an example of a VINCA interaction specification. The message exchanged between users and services is also illustrated here. The same message will be delivered to different channels in suitable formats, such as cHtml, HTML and WML, according to the interaction specification, resulting different effects.


Message Exchanged
Adapt to different interaction channels
StarCaddy
Notification
The Riding event had been postponed because of the heavy rainfall!
You can click ok button to get your new schedule.
ok
back

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseMsg>
    <Message title="Notification">
        <Description>
The Riding event had been postponed because of the heavy rainfall! You can click ok button to get your new schedule.
        </Description>
    </Message>
</ResponseMsg>
```

Fig. 3.4. Multiple Access Channels

3.3 The Supporting Environment of VINCA

We have implemented a prototype environment to support power users in building Web-based applications with VINCA. The system architecture of the supporting environment is shown in Fig.3.5.

The VINCA Visual Programming Environment enables power users to express their requirements from business viewpoint in a WYSIWYG (What You See Is What You Get) manner. Its output is the so-called VINCA program – a set of specifications in XML. Then, the VINCA Interpreter will transform them into BPEL4WS-alike executable specifications.

The BPEL4WS-based Running Environment is responsible for chaining Web-based services at operational level, and binding and invoking individual services. It consists of an event manager, an application scheduler, a location manager, a time manager and a process engine [Holt03].

3.4 Application

Fig.3.6 is a snapshot of a power user’s working interface supported by the business-end programming environment. It is based on the simplified scenario discussed in Section 3.1. John or his secretary can take the role of a power user and sketch his schedule during the Olympic Games. When he arrives at Beijing, his application is triggered. Whenever he changes his mind or his predefined schedule is influenced by some unexpected factors, John can reconstruct his personal application in a just-in-time manner through directly composing business services made available to him through the service community.

A third-party agent, such as the Olympiad organizing committee, can define some common-used processes as value-added composite services (see the upper-right corner of Fig.3.6). Since these pre-defined composite services normally cannot 100-percent satisfy diverse requirements of different users; John as a power user can customize and extend his personalized service composite based on pre-defined composite services as well as basic business services available to him. The customized process is shown in the lower part of Fig.3.6.

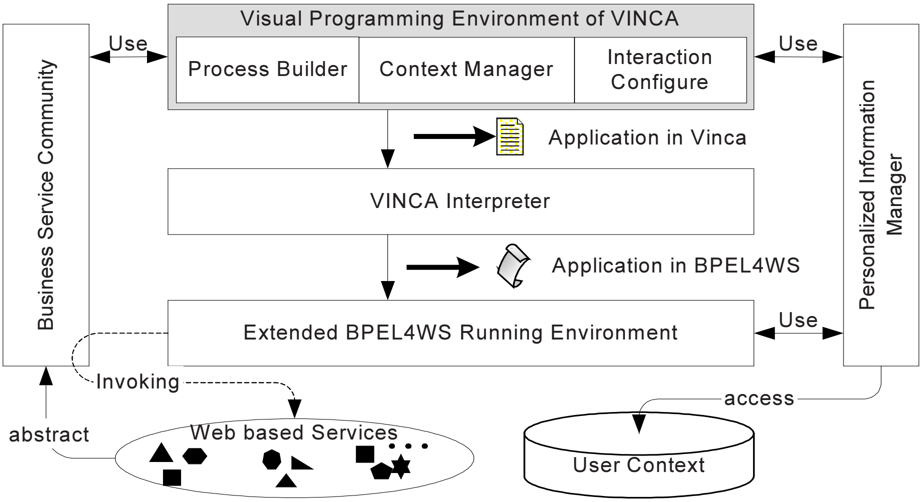


Fig. 3.5. The Supporting Environment of VINCA

Using the running example, we sum up how the VINCA-based approach works with the following major steps:

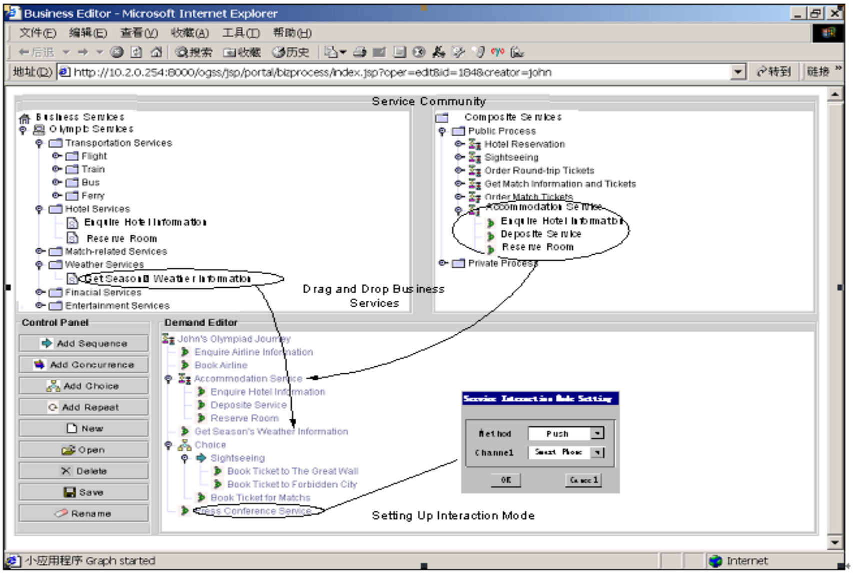


Fig. 3.6. Snap-shot of VINCA Programming Environment

Setting up a business program skeleton. The VINCA programming environment provides an intuitive way to use potential templates for setting up business program skeletons. Templates are actually also service composites maintained by the system and organized according to domain-specific classifications. With the help of templates, John does not need to start from scratch for typical use cases.

Checking and locating appropriate business services. For each power user, the VINCA programming environment filters out all available business services in terms of the privileges and corresponding user context of the user, and organizes them according to a domain-specific classification.

Completing the business program. After locating his desired services, John can drag and drop these services into the Business Program area, and use the control elements listed on the left side to glue them together. John can either explicitly specify the input-output relationships of interrelated services or leave them to be determined at runtime.

Defining the interaction modes. After getting the business program, John configures the interaction mode for his application. Then he can get the expected results in the way he defined in the interaction specification.

4 Conclusion

The central idea of the reported research is to let users “see” all the network-based resources available to them and “program” their personalized applications by configuring their resources in a straightforward manner using business-related concepts only. Based on the integrated model that relates some key elements of a business-level design issues and their counterparts in the software domain in a convergent way, our approach promotes the metaphor of business-end programming and enables demand-driven and user-centric configuration of service-based applications. In the context of the FLAME2008 project, the first working version of VINCA is proposed and implemented. The prototypical implementation is quite promising and attracts a lot of attention from interested parties. Research and development are underway, and a full-fledged implementation is expected early 2005.

References

[Abow00] A.K. Dey, G.D. Abowd, Towards A Better Understanding of Context and Context-Awareness, ACM Conference on Human Factors in Computer Systems (CHI 2000), The Hague, Netherlands, April 2000

[Andr03] T. Andrews, et al, Business Process Execution Language for Web Services Version 1.1, http://www-106.ibm.com/developerworks /ws-bpel/, May 2003

[Brow97] P. J. Brown, J. D. Bovey, and X. Chen, Context-Aware Applications: from the Laboratory to the Marketplace, IEEE Personal Communications,Vol.4, No.5,1997

[Cafi02] CAFISE group, Service Community Specification, Technical Report, Software Division, ICT of CAS, December 2002

[Casa01] F. Casati, M. Sayal, M. Shan, Developing E-Services for Composing E-Services, Advanced Information Systems Engineering: 13th International Conference, CAiSE 2001, Interlaken, Switzerland, June 4–8, 2001

[Han03] Y. Han, et al, CAFISE: An Approach Enabling On-Demand Configuration of Service Grid Applications, Journal of Computer Science and Technology, Vol.18, No.4, 2003

[Holt03] B. Holtkamp, R. Gartmann, Y. Han, FLAME2008-Personalized Web Services for the Olympic Games 2008 in Beijing, Proceedings of eChallenges 2003, Bologna, Italy, Oct. 2003 (to appear)

[Kici01] E. Kiciman, L.L. Melloul, A. Fox, Towards Zero-Code Service Composition, Proceedings of the Eighth Workshop in Hot Topics in Operating Systems (HotOS VIII), 2001

[Krit01] R. Krithivasan, A. Helal, BizBuilder- An e-Services Framework Targeted for Internet Workflow, Proceedings of the third Workshop on Technologies for E-Services (Tes'01), Springer lecture notes in Computer Science series, Vol.2193

[Picc02] G. Piccinelli, S. L. Williams, Workflow: A Language for Composing Web Services, Second International Workshop on Composition Languages In conjunction with 16th European Conference on Object-Oriented Programming (ECOOP), Málaga, Spain June 11, 2002

[Shen02] Q. Sheng, B.Benatallah, M.Dumas, E. Mak, SELF-SERV: A Platform for Rapid Composition of Web Services in a Peer-to-Peer Environment, VLDB 2002

[Sing01] M. P. Singh, Physics of Service Composition, IEEE Internet Computing, Vol.5, No.3, May/June 2001

[Weis01] J. B. Weissman, B. Lee. The Service Grid: Supporting Scalable Heterogeneous Services in Wide-Area Networks, Proceedings of Symposium on Applications and the Internet, January 2001

[Weis03] N.Weissenberg, R.Gartmann, An Ontology Architecture for Semantic Geo Services for Olympia 2008, in: IfGIprints 18, Proc. GI-Tage , Münster, June, 2003