ABSTRACT
Facing the problems of existing teleconsultation platform, this paper puts a notion of constructing a service-oriented and event-driven teleconsultation platform, in order to solve the resource-sharing and cooperative-working related problems which medical institutions will meet during the buildup of dynamic alliance. Consultation event is used to express the requisites of consultations between medical institutions and to support semantic exchange between Hospital Information Systems. The publish/subscribe mechanism of topic and content based event is described, and partner discovery and selection algorithms are given thereafter. On the base of our research, a prototype project of the platform, named as TMCGrid, has been implemented.

KEY WORDS
Telemedicine, teleconsultation, event-driven, partner selection

1. Introduction
Service-oriented architecture (SOA) is a new integration approach of distributed applications. The framework of SOA consists of three roles and three primary operations. Three roles are service provider, service requestor and service broker, while three operations are publishing, finding and binding [1].

1.1. Shortage of centralized system
At the present time, the architecture of most teleconsultation platforms is centralized. This kind of platform provides service to hospitals locate in the same geographic region with its data center individually. Poor network condition often leads to low quality of communication, and failure of data center causes all service to terminate. Therefore, distributed teleconsultation platform is considered and will become a trend undoubtedly.

1.2. Lack of cooperation
Although the number of telemedicine center has increased a lot during the past several years, the service scope of each one is limited and the cooperation between them is little, which prohibited the development of telemedicine and also resulted in redundant construction. It can be expected that the requirement of cooperation in consultation and resource sharing will become more notable, and the application of cross-region or even cross-national consultation will be more frequently in future.

1.3. Approach of seeking partner
Because of collectivizing of medical treatment fields, competition between medical institutions has become more and more drastic. Facing these situations, hospitals would like to establish some alliances based on consultation cooperation, in order to earn from other’s strong points to offset their weakness and improve their competitive ability.

Facing these problems, we put a notion of constructing a service-oriented and event-driven teleconsultation platform. Consultation event is used to express the requisites of consultations between medical institutions and to support semantic exchange between Hospital Information Systems. The publish/subscribe mechanism of topic and content based event is described, and partner discovery and selection algorithms are given thereafter, in order to solve the resource-sharing and cooperative-working related problems which medical institutions will meet during the buildup of dynamic alliance. On the base of our research, a prototype project of the platform, named as TMCGrid, has also been implemented.

2. Constructing Event-Driven Consultation Alliance
The core in the approach of constructing an event-driven consultation cooperative alliance is consultation event and its processing mechanism. Consultation events are used to capture specific operations, such as request of consultation in HIS. It also includes data to be exchanged during the whole process, such as case record, preliminary diagnose, consultation orientation, etc. Consultation event templates are used to describe the regulation of topic and content in events. Hospitals can use event templates to configure
specific operations and data to be captured during consultation. The mechanism of processing publish/subscribe based consultation event includes five core objects and three kinds of operations. Five core objects are consultation event template, consultation requestor, consultation provider, subscription event and event broker. Three kinds of operations are event publish, event subscribe and process of event broker [2]. Figure 1 indicates the relationship between there objects and operations.

![Figure 1. The mechanism of processing publish/subscribe based consultation event](image)

2.1. Consultation event and its processing mechanism

During the process of consultation event, requestor publishes consultation events in order to seek partners. Local event broker will handle the request events and search for the proper subscribers in the subscription event database, which includes subscription events from consultation providers in both local and remote regions. For consultation providers in remote region, subscription events are sent to local event broker by remote event broker, which works as subscribe agent. After the proper providers are found, requestor can negotiate with providers and tries to set up a cooperative alliance for current consultation [2].

2.1.1. Definition of consultation event and consultation event template

Definition 1. $ET = (Topic, Content)$ is 2-tuple describing data structure of each consultation event template. Component $Content$ is a set of $Description$, which is a 3-tuple: $(Name, Semantic, Value)$.

Definition 2. $E = (ID, Topic, Type, Content, Source, Destination, DateTime)$ is 7-tuple describing data structure of consultation event. Components of $E$ represent the unique identity, topic, type, content, source, destination, generating time of the event in sequence. $Source = (Requestor, Org, CU)$ is 3-tuple describing doctor, medical institution (including department, hospital, telemedicine center), current consultation alliance where the event is sent from. $Destination = (Subscriber, Org, CU)$ is 3-tuple describing doctor, medical institution, current consultation alliance where the event is sent to. Consultation event is instantiated from consultation event template.

2.1.2. Analysis of one-to-one cooperation

A consultation requestor will receive result, saying successful or failed, from a consultation provider each time it sends a request for cooperation. The whole process of such a request is called one-to-one cooperation, which includes three stages, seeking provider, negotiating with provider and executing request. During one-to-one cooperation, many events are involved. The relationship between them is described as figure 2. As shown by the figure, several round of cooperation negotiation (T2) is required by consultation requestor and consultation provider to make terms, and the process of service invoke (T5) and response (T6) may also be executed repeatedly. If there is no subscriber (T8), the cooperation is rejected (T4), or unrecoverable error occurs during service invoke, this round of one-to-one cooperation will fail and has to be restarted after being adjusted.
2.1.3. Event processing during life cycle of consultation alliance

Sometimes, one-to-one cooperation will become deficient. This is because of lack of equipment or technique of consultation provider, or some diseases with higher difficulties and complexities. In the case of that, the consultation event should be divided in order to seek several consultation providers to perform the consultation together. This is called one-to-many consultation, which is based on one-to-one consultation but far more complex than it.

2.2. Subscription event and consultation alliance

**Definition 3.** $S = (ID, Application, Topic, Conditions)$ is a 4-tuple describing data structure of subscription event. Components of $S$ represent unique identity, consultation application, topic of consultation event, conditions that restrain contents of consultation event in sequence. $Conditions = \{Condition\}$, $Condition = (Name, Operator, Value)$ describes a relation expression.

During the process of consultation cooperation, the management of consultation event templates and subscription events is very important. A common space, called consultation operation community, is required to store these data.

**Definition 4.** $OC = (ID, Criterions, ETs, Orgs, OP)$ is a 5-tuple describing data structure of consultation union. $Criterions = \{Criterion\}$ describes operation criterions. $ETs = \{ET\}$ is a set of event templates. $Orgs = \{Org\}$ records all the medical institutions in the operation community. $Org = (Name, Type, Applications, \{S\}, OrgOP)$ is a 5-tuple, $Name$ represents the name of institution, $Type = \{Hospital, Department\}$ represents the type of institution, $Applications = \{Application\}$ is a set of applications refer to the institution, $\{S\}$ is a set of subscription events, $OrgOP = \{Subscribe, Unsubscribe\}$ includes event subscribe/unsubscribe operations. $OP = \{Register, Unregister\}$ includes register/unregister operations of Criterion, ET and Org.

Consultation operation community is owned and administered by telemedicine center. The relationship between consultation operation communities and telemedicine centers is many-to-one. If a medical institution in a community publishes a consultation request, partner will be found by looking up subscription events, and consultation alliance will be build up after negotiation.

**Definition 5.** $CU = (ID, Services, Relations, EventChain, OP)$ is a 5-tuple describing the data structure of consultation alliance. $Services = \{Service\}$ describing services in the alliance. $Relations = \{(Partner, Service)\}$ is a set of relationship between partner and service, in which $Partner \in OC.Orgs$. $EventChain = (Events, Relation)$ describes the event chain generated during the process of consultation. $OP = \{addPartner, addEvent\}$ includes operations to add partner or event into the alliance.

Events in consultation alliance are sequential and cross-referenced. After the occurrence of consultation request event, other events will be touched off and an event chain will be established. As different events are corresponding to different services, these services are invoked by the sequence of events in the chain, and results are returned. Thereby, the workflow of the consultation alliance does run [2].

2.3. Discovery and selection of consultation partner

2.3.1. Matching relation between consultation event and subscription event

**Definition 6.** Topic match: If the topic of consultation event $e$ consists with or consists in the topic of subscription event $s$, then $e$ and $s$ is topic match, marked as $e.Topic \subseteq s.Topic$.

**Definition 7.** Content match: For $s.\text{Conditions} = \{\text{Condition}_1, \text{Condition}_2, \ldots, \text{Condition}_m\}$, $e.\text{Content} = \{\text{Description}_1, \text{Description}_2, \ldots, \text{Description}_n\}$, $m \leq n$, if $\forall \text{Condition}_i \in s.\text{Conditions}$ ($1 \leq i \leq m$), $\exists \text{Description}_j \in$
e.Content (1≤i≤n): (Description, Name = Condition, Name) \(\wedge (\text{Condition, Operator}(\text{Description}, \text{Name}) = \text{Condition, Value})\), then e and s is content match, marked as e.Content \(\subseteq s.CConditions\).

**Definition 8.** Event matching relation: According to the matching degree between e and s, the matching relation between them has three types [2].

1) If \(\forall e, \exists s\): (e.Topic \(\subseteq s.Topic\)) \(\wedge\) (e.Content \(\subseteq s.CConditions\)), then e and s is totally match, marked as e\(\rightarrow s\).
2) If \(\forall e, \exists s\): (e.Topic \(\subseteq s.Topic\)) \(\wedge\) (e.Content \(\subseteq s.CConditions\)), then e and s is partially match;
3) If \(\forall e, \not\exists s\): (e.Topic \(\subseteq s.Topic\)), then e and s is not match, marked as e\(\rightarrow \phi\).

**Definition 9.** Event is dividable: When e and s is partially match, if \(\exists e_1, e_2, \ldots, e_n, s_1, s_2, \ldots, s_m\): (e\(_1\).Topic = e\(_2\).Topic = \ldots = e\(_n\).Topic \(\subseteq s_.Topic\)) \(\wedge\) (e\(_1\).Content \(\subseteq s_1.CConditions\)) \(\wedge\) (e\(_2\).Content \(\subseteq s_2.CConditions\)) \(\wedge\) \(\ldots\) \(\wedge\) (e\(_n\).Content \(\subseteq s_m.CConditions\)) \(\wedge\) (e\(_1\).Content \(\cup\) e\(_2\).Content \(\cup\) \ldots \(\cup\) e\(_n\).Content) = e.Content) \(\wedge\) ((e\(_1\).Content \(\cap\) e\(_2\).Content \(\cap\) \ldots \(\cap\) e\(_n\).Content) = \(\phi\)), then e can be divide into sub-events e\(_1\), e\(_2\), \ldots, e\(_n\), marked as e\(\rightarrow\{e_1, e_2, \ldots, e_n\}\) \(\rightarrow\{s_1, s_2, \ldots, s_m\}\).

2.3.2. Subscription based consultation partner discovery

Subscription based consultation partner discovery provide a reliable method to find out the proper subscribers according to the conditions from a large amount of subscription events. We may use the following match evaluation function to calculate the partner candidates we want.

\[ f_{Mart} (e) = \begin{cases} \{s\}, e \rightarrow s \\ \{s_1, s_2, \ldots, s_m\}, e \rightarrow \{e_1, e_2, \ldots, e_m\} \rightarrow \{s_1, s_2, \ldots, s_m\} \\ \phi, e \rightarrow \phi \end{cases} \]  

2.3.3. Consultation partner selection algorithm

Many partner candidates are found during the seeking process, but only one group will be selected for a consultation. The problem is how to select the best group of providers and invite their services [3]. Our algorithm is based on the estimating index for hospital (\(W_{\text{Hospital}}\)) and the estimating index for department (\(W_{\text{Department}}\)).

**Definition 10.** \(W = (\text{OrgName, OrgType, Weight, \{OP\}}\) is a 4-tuple describing the data structure of estimating index. Components of \(W\) represent the name of the institution, the type of the institution, the estimating index of the institution and the set of operation in sequence. \(OP = \{\text{addOrg, removeOrg, updateOrg}\}\) describes the operation including insert, delete and update.

The factors of estimating index for hospital are Quality (Q), Level (L), Creditability (C), Price (P) and Miscellaneous (M). Many guidelines should be determined by expert group. Quantifying regulations of these factors are \(F_p, F_q, F_c, F_m\). qualified standards of them are \(Q_1, L_1, C_1, P_1, M_1\), and weight percentages of them are \(q\%, l\%, c\%, p\%, m\%\). They match the formula \(0\leq q\%, l\%, c\%, p\%, m\%\leq1\) and \(q\%+l\%+c\%+p\%+m\%=1\). We use the following function to estimate whether a factor is qualified or not.

\[ f_{\text{Verify}}(\text{Value, Threshold}) = \begin{cases} \text{Value, Value } \geq \text{ Threshold} \\ 0, \quad \text{Value < Threshold} \end{cases} \]  

Calculating the function \(f_{\text{Verify}}(\text{Value, Threshold})\) with \(Q, L, C, P, M\) as Value and \(Q_1, L_1, C_1, P_1, M_1\) as Threshold, we will get \(Q', L', C', P', M'\). The formula of estimating index for hospital is:

\[ W_{\text{Hospital}} = (Q*q\% + L*l\% + C*c\% + P*p\% + M*m\% \)  

On the base of estimating index of hospital, the factors of estimating index for department are Skill (S), Quality (Q), Creditability (C) and Miscellaneous (M). Quantifying regulations of these factors are \(F_s, F_q, F_c, F_m\). qualified standards of them are \(S_1, Q_1, C_1, M_1\), and weight percentages of them are \(s\%, q\%, c\%, m\%\). They also match the formula \(0\leq s\%, q\%, c\%, m\%\leq1\) and \(s\%+q\%+c\%+m\%=1\). Calculating the function \(f_{\text{Verify}}(\text{Value, Threshold})\) with \(S, Q, C, M\) as Value and \(S_1, Q_1, C_1, M_1\) as Threshold, we will get \(S', Q', C', M'\). The formula of estimating index for department is:

\[ W_{\text{Department}} = W_{\text{Hospital}} + \left( S*s\% + Q*q\% + C*c\% + M*m\% \right) ; \]  

\[ \left( S'*Q'*C'*M' \right) \times \frac{W_{\text{Hospital}}}{W_{\text{Hospital}}} \]  

\[ W_{\text{Hospital}} = \frac{1}{n} \sum_{i=1}^{n} \left( W_{\text{Hospital}} \right) \]  

In eq. (4), \(W_{\text{Hospital}}\) is the estimating index of the hospital which the department refers to, \(W_{\text{Hospital}}\) is the average of the estimating indexes of all the hospitals in the consultation operation community. The evaluation function for estimating medical index is as following.

\[ f_w(\text{Org}) = \begin{cases} W_{\text{Hospital}}(\text{Org}), \quad \text{Org.Type} = \text{Hospital} \\ W_{\text{Department}}(\text{Org}), \quad \text{Org.Type} = \text{Department} \end{cases} \]  

The following is our consultation partner selection algorithm.
Input: Event e, Consultation operation community OC.

Step 1. Seek all the topic match subscription events in OC, compose a set \( SC = \{ s \mid (e.\text{Topic} \subseteq s.\text{Topic}) \wedge (s.\text{Org} \subseteq \{ \text{Orgs} \}) \} \). Let \( S = \phi \), \( P = \phi \).

Step 2. Calculate relation \( S_\text{Org} = \{ (s,\text{Org},w) \mid (s \in \text{SC}) \wedge (w = f_\text{org}(\text{Org})) \wedge (f_\text{org}(\text{Org}) > 0) \wedge (s.\text{Application} \subseteq \text{Org}.\text{Applications}) \wedge (\text{Org} \subseteq \text{OC}.\text{Orgs}) \} \).

Step 3. Sort relation \( S_\text{Org} \) by field \( w \) descendingly.

Step 4. Selection while event totally match:

for each \( s \_\text{org} \) in \( S_\text{Org} \)

\[ S \leftarrow \pi_s(s \_\text{org}), P \leftarrow \pi_{\text{org}}(s \_\text{org}) \]

if e.\text{Content} \subseteq \text{SCConditions}

if negotiate with P successfully

exit

end if

end for

Step 5. Use event division algorithm to divide event \( e \) into sub-events \( e_1, e_2, \ldots, e_m \) satisfying that \( e \rightarrow \{ e_1, e_2, \ldots, e_m \} \rightarrow \{ s_1, s_2, \ldots, s_m \} \). Mark \( \{ s_1, s_2, \ldots, s_m \} \) as \( S_g \), set \( \{ S_g \} \) includes every solutions of \( S_g \).

Step 6. Define relation \( S_\text{G}_\text{Org} \) as set of 3-tuple \( (S_g, P_g, W_g) \), \( S_\text{Org} = \phi \);

for each \( s_g \) in \( \{ S_g \} \)

\[ w_g = \left( \sum_{i=1}^{m} \frac{\pi_w(\sigma_{s=s_i}(S_\text{G}_\text{Org}))}{m} \right), \]

\[ s_i \in s_g = \{ s_1, s_2, \ldots, s_m \} \]

\[ pg = \pi_{\text{org}}\left( \sigma_{s=s_i}(S_\text{G}_\text{Org}) \right) \]

\[ s_i \in s_g = \{ s_1, s_2, \ldots, s_m \} \]

\( S_\text{G}_\text{Org} \leftarrow S_\text{G}_\text{Org} \cup \{ (s_g, pg, wg) \} \)

end for

Step 7. Sort relation \( S_\text{G}_\text{Org} \) by field \( W_g \) descendingly.

Step 8. Selection while event partially match:

for each \( s_\text{org} \) in \( S_\text{G}_\text{Org} \)

\[ S \leftarrow \pi_{s_g}(s_\text{org}), P \leftarrow \pi_{s_g}(s_\text{org}) \]

for each \( p \) in \( P \)

if negotiate with \( p \) successfully

if negotiate with \( p \) failed

\[ \text{co_reject: exit} \]

end if

end for

end for

Output: \( S \) and \( P \), which the algorithm gives, are sets including selected subscription events and consultation partners. If the output of the algorithm is empty set, it means that no proper partner is found and manual interference is required to adjust the consultation apply event.

3. Key Problems of Service-Oriented Teleconsultation Platform

3.1. Migrate from existing platform

Although the implementation of former teleconsultation platforms varies a lot, the major services provided by them and the workflow of them are similar. All of them have primary functions such as member registration and management, consultation application and arrangement, financial management, etc. This gives the opportunities for these platforms to share resources and work cooperatively. Meanwhile, the modification of existing system can be reduced as little as possible.

3.2. Description of data

The major data description criterion existing in medical information fields is Health Level Seven (HL7) Standards [4], which is based on Extensible Markup Language (XML). As a service-oriented teleconsultation platform, TMCGrid describes most of its data with XML, and uses SOAP, an XML based protocol, to communicate between services. As the patterns of all these data can be defined by DTD or XML Schema, the capability and extensibility of the system is ensured.

3.3. Integrate with existing resource and equipment

The former teleconsultation platform was commonly designed on B/S architecture, which made it hard to share data directly with other B/S or C/S based HIS, PACS, etc [5]. On the contrary, TMCGrid is a system which is based on J2EE platform and service-oriented architecture; it uses Web services as its interfaces to communicate with other systems, and adopts Struts framework, MVC (Model-View-Control) developing mode and reusable component technology. All these technologies help the platform improve its extensibility and flexibility in integrating with existing resource and equipment.

3.4. Security

During the process of teleconsultation, much information involving patients’ privacy is stored and transmitted. This calls for high security of the platform, including trust relationship between services and data encryption during transmission. TMCGrid adopts a Public Key Infrastructure (PKI) based X.509 certification authentication mechanism, Secure Socket Layer(SSL) and WS-Security[6] based SOAP messaging enhancement to ensure the security of data [7].

3.5. Authentication of users

Each telemedicine center has its own user management method, the account information is owned by the telemedicine center the user refers to. When the user wants to access the services provided by other telemedicine center,
as his identity can’t be recognized, the stranger problem occurs. In order to solve these kinds of problems, TMCGrid uses client certification based one site login mechanism for authentication of users.

4. Implementation of Teleconsultation Platform

4.1. Design and development of TMCGrid system

As a multi-tier service-oriented application, TMCGrid is J2EE-based. It consists of several major modules, including TMC_Data, TMC_Base, TMC_WebService, TMC_WebServiceClient, TMC(WebApp), etc.

- **TMC_Data**: The major function of this module is to encapsulate data access tier. Hibernate, which makes O/R mapping and data persistence much easier, is used to simplify database access operations. With Hibernate, TMCGrid can be easily migrated to any relational database.

- **TMC_Base**: As the core of the whole system, this module includes implementation details of each function such as hospital management, user management, consultation establishment, partner seeking and selection, etc. It provides APIs which has specific functions to other modules.

- **TMC_WebService and TMC_WebServiceClient**: These two modules encapsulate APIs which are exposed to other systems such as HIS or other teleconsultation platform. It realizes the interoperability between systems in different architecture.

- **TMC(WebApp)**: This module is a web tier application, which adopts Struts framework. It provides UI for doctors as well as administrators to perform variety kind of operations.

4.2. Testing scenario for TMCGrid

In order to test TMCGrid, we’ve established a simulative environment, which includes several servers with proper software (Linux FC3, JDK 1.5, Tomcat 5.0) installed and several PCs. Servers are divided into three groups to simulate three telemedicine centers, each center has several member hospitals.

Firstly, we performed some test on each group separately. This seemed to be very similar to the original centralized teleconsultation platform. The different was that partner discovery and selection during each group no longer require manual interference.

Secondly, a cross-group test was performed. This process verified the data interchange and cooperation between different telemedicine centers works well. Meanwhile, we assumed several samples of estimating index for each hospital and department in these three groups, and found out that the partner selection results were satisfying.

5. Related Works and Problems

During our research, we’ve also found some problems, which should be solved in our further work: (a) Establish general consultation event templates. A serial of canonical event templates are the base of the platform and they can help the platform to be more extensible. (b) Determine quantifying regulations on the factors of estimating index for medical institutions. (c) Integrating with HIS and PACS. As the former HIS, PACS adopts some technologies such as RPC, CORBA to support distributed specification, some improvement should be make to improve the interoperability between TMCGrid and these systems.

6. Conclusion

The construction of service-oriented and event-driven teleconsultation platform is a huge project. Designing a feasible and stable prototype and establishing criterions based on the prototype are preconditions of the whole project. TMCGrid gives a good implementation, but further research and practice are still required before it can be put into service.

Acknowledgement

This research was supposed by China Medical Board Research Fund (CMB, 99-701). The authors would like to acknowledge Zuo Mei-Qing, Xia Zhi-Yuan, Luo Xiao-Zhen, Chen Xu-Guang for their assistance in the developing and testing works of TMCGrid, and also acknowledge Affiliated Huashan Hospital of Fudan University for supporting on this research.

References

[5] Hongyan Li; Ming Xue; Ying Ying, A Web-based and integrated hospital information system, Proceedings of the IDEAS Workshop on Medical Information Systems, The Digital Hospital (IDEAS-DH’04), Beijing, P.R.C, 2004, 157-162