UDC 004.412

A. SYROTYUK

National Aerospace University Named After M.E. Zhukovsky, Ukraine

ARCHITECTURE AND IMPLEMENTING SERVICE-ORIENTED PHARMACOLOGICAL SYSTEM "MEDICINE"

Present work is devoted to service-oriented systems problems. Particularly, medical web-services are discussed. Information technologies classification in medicine and urgency of web-services using in medicine are formulated. Unique service-oriented pharmacological system development is proposed. The making goal, architecture of given system and its functions are described. Interconnection of system functions and medicine data is given as schema. Operations of logical derivations which are determinated on clear rules based on the analysis of Boolean matrixes are proposed for implementation of given interconnection. Designing any web-service one should takes into account its dependability. Therefore the analysis of web-services dependability intermediums. Dependable service-oriented architecture and system diversity models, provided its dependability, were developed. Experimental and also further searches of service-oriented pharmacological system "Medicine" are considered.

medicines web-service dependability

Introduction

Drug therapy complications problem is getting more and more urgent all over the world. It was found out in special researchers that millions of people suffered from serious, sometimes even irreversible, complications, got as a result of drug therapy.

A doctor should realize his responsibility to the patients, who will take prescribed medicine. Despite this, a percentage of doctor's mistakes in drug practice is quite high. Among doctor's mistakes types there were detected such as drug intolerance, unjustified drug choice, dosage mistakes, simultaneous use of two or more drugs of different groups without taking into account their interaction, etc [1, 2].

However, it's very important that negative side reactions are potentially avertible. One of the most effective ways to encrease rational prescribing is information technologies application to medicine and providing access from both sides of spicialists and patients.

On the base of hold analysis of IT implementation into public health system existed computerization elements classification in the sphere of medicine was formulated. Since the Internet has become one of the most important sourses of relevant professional information for medical community and as a revolutional communicational technology has opened new ways of individual group interaction, telemedicine was recognized as one of the most multifunctional and effective interaction between information technologies and medicine [3]. Respectively, nowadays telemedicine has a numerous of constituents, which execute various functions. These constituents are web-services, which can be provided by either, one or another organisation or company into public use.

Web Services Model Providing the Healthcare Industry

So, what is a "Web-service" you will ask. According to the W3C a Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface that is described in a machine-processable format such as WSDL [4].

The Web services model provides the healthcare industry with an ideal platform to achieve the difficult interoperability problems. Web services are designed to wrap and expose existing resources and provide interoperability among diverse applications [5]. Numerous amounts of medical web-services one can classify as follows: e-databases, e-market of medicaments, expert support systems, telemedicine, etc.

Pharmacological Web Service for Prescriptions Verifying

Although there is a great deal of medical web-services in the Internet, field of their application is much wider. And, of course, these services cover not all needs present-day client requires. Everyone knows we look to health IT to improve quality by making the patient's information available. But we also look to health IT to make the best treatment information available.

A very good idea appeared concerning this problem. We've decided to develop a service-oriented pharmacological system called "Medicine" that will help physicians and patients in verifying their prescriptions. This includes correspondence analysis to the disease, verification of medicine compatibility, contraindications and drug side effect verification, taking into account individual physiological patient features (age, sex, pregnancy, chronic diseases, etc.). In that way, we hope that it will be fewer mistakes made by doctors and that it will be more awareness from the direction of people whom these medicines have been prescribed.

To achieve those ends, we propose our "Medicine" system architecture. Its architecture is described in [6].

As it submitted above, the "Medicine" is an expert support system. This web-service has no proper analogues. Thus, this system is unique and extremely useful. We believe that the web-service will help physicians to prescribe medicine properly without any mistakes, that is especially important to people who have different chronicle diseases, to children, pregnant women, etc.

To be able to provide full information about medicine it's necessary to design database itself.First step is to design logical and physical models of our pharmacological Database.

Second one is to design DataBase module structure.

Third, we should find information about medicine (from paper or electronic pharmaceutical manuals) and, finally, to fill our Database with appropriate medicine information.

Considering goals of this system design, web-service functionality was developed. Thereby, our service will provide:

- full information about medicine;
- two or more prescribed medicine verification;
- medicine and patient's chronic disease verification;
- medicine and treated disease verification;
- medicine and particular patient's state verification.

A very important question is which methods to use to compare medicine, diseases, etc.

We've chosen operations of logical derivations which are determinated on clear rules based on the analysis of Boolean matrixes. You can observe these matrixes below (tabl. 1).

Table 1

Medicine and patient's chronic disease comparison

Chronic / Medicine	Renal insuffi ciency	Liver deseases	Hypersinsitivity to medicine components	Stomach ulcer
ACC	-	+	-	+
roksid	+	-	-	+
sempreks	-	+	-	+
travisil	+	+	-	+

The above matrixes are supposed to be used in an experimental model of the system. We are planning to extend these operations by input of additional fuzzy logic rules since points of medicine prescriptions usually are not all really strict.

Web Services Dependable Architecture Model

What is quite clear in designing and implementing web-services is that they should be dependable. Achieving high dependability of Web-service is still an open problem. And although our pharmacological system is not a critical one, the problems of its dependability are being discussed. An important goal is to provide a dependable web-service. For that reason a diversity arcitecure of our system is proposed [7, 8].

Web-services high dependability and fault-tolerance are ensured by using various kinds of redundancy and diversity on different system arcitecture levels.

Further a dependable architecture model of SOPhS is introduced. This model describes diversity versions for initial service-oriented system (fig. 1). Full and partial replacement to diversity system are proposed on tha base of data about vulnerabilities of diversity levels elements [10]. Such data can be obtaineed from National Vulnerability Database (NVD). One of the main parts of the aricitedcture is a Configuration Control Server, which provides partial or full system reconfiguration. For example, if meaningful vulnerabilties in MySQL DBMS are discovered, then a system will use another DBMS (e.g. Oracle). It will last until configuration with the chosen DBMS considered to be at most dependable.

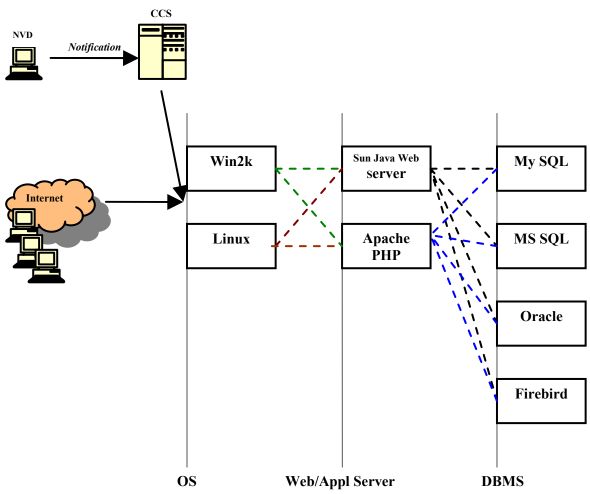

Fig. 1. A dependable SOPhS architecture

Having analyzed system diversity architecture, we can make a conclusion, that there several versions of partial or full system diversification.

If given architecture is simplified, diversity models will be obtained, using wich a comparative analysis of the whole diversity models quantity will be made easier. In diversity models replacement redundancy principles are used, where a reserve is considered to be cool (inactive).

Here is a probability of no-failure comparative analysis of initial and diversity system models.

First of all, we have determined PNF [9] for inital SOPhS (fig. 2, form.1-2):

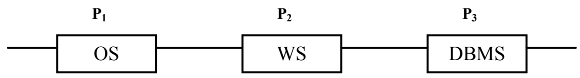

Fig. 2. Initial SOPhS model

$$P_{CO\Phi C} = P_1 \cdot P_2 \cdot P_3; \tag{1}$$

$$P_{CO\Phi C} = e^{-\lambda_{os}t} \cdot e^{-\lambda_{ws}t} \cdot e^{-\lambda_{db}t} = e^{-t(\lambda_{os}+\lambda_{ws}+\lambda_{db})}. \tag{2}$$

The first model takes into account diversion only on the Operating System level (fig. 3).

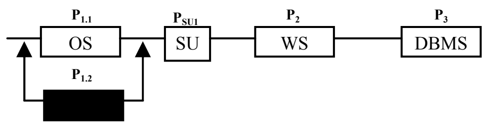

Fig. 3. Diversion on the OS level

PNF of first model is:

$$P_{m1} = P_1 \cdot P_2 \cdot P_3, \tag{3}$$

where
$$P_1 = e^{-\lambda t} \cdot (1 + \lambda t) \cdot P_{SU1}; \tag{4}$$

P_{SU} – PNF of switching unit, which is responsible for system reconfiguration on the definite diversion level.

Here we can neglect it, as its PNF is three degrees higher than diversion level's ones. Then,

$$P_{m1} = e^{-\lambda_{os}t} \cdot (1 + \lambda_{os}t) \cdot e^{-\lambda_{ws}t} \cdot e^{-\lambda_{db}t}; \tag{5}$$

$$P_{m1} = e^{-t(\lambda_{os}+\lambda_{ws}+\lambda_{db})} \cdot (1 + \lambda_{os}t). \tag{6}$$

In the same way PNF of other models have been formulated. The second model takes into account diversion only on the Web Server level (fig. 4)

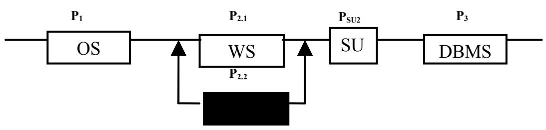

Fig. 4. Diversion on the WS level

PNF of the second model is

$$P_{m2} = e^{-t(\lambda_{os}+\lambda_{ws}+\lambda_{db})} \cdot (1 + \lambda_{ws}t). \tag{7}$$

The third model takes into account diversion only on the DatBase Management System level (fig. 5).

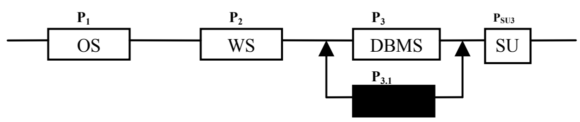

Fig. 5. Diversion on the DBMS

PNF of third model is

$$P_{m3} = e^{-t(\lambda_{os}+\lambda_{ws}+\lambda_{db})} \cdot (1+\lambda_{db}t) . \qquad (8)$$

The fourth model takes into account diversion on OS and WS levels. Here we have viewed two cases.

First configuration of the fourth model implies one Switching Unit, giving orders to switch mode Operating System – Web Server (fig. 6).

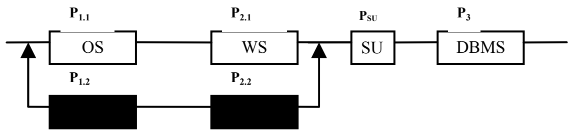

Fig. 6. First configuration of the fourth model

Second configuration of the fourth model implies two Switching Units. Each of them is assigned to control its own level (fig. 7). Suc configuration is considered to be more reliable than the first one. In case, if one of the SU is failed, the second SU won’t and the system remains capable for reconfiguration, although not full. In first case, when the oly SU is failed, the system can’t use diversification.

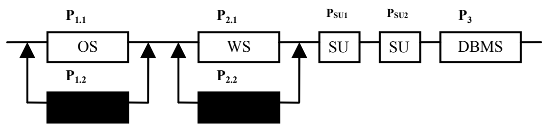

Fig. 7. Second configuration of the fourth model

Of course, adding extra SU, in calculations PNF of system should be lower. But as PNF of SU is several degrees lower than SU of diversity elements, and at the same time in practice system reliability encreases much, its more rationally to use the model with exactly separate diversion replacement, using thus several Switching Units. Therefore, in this case a model, depicted in fig.10, is not relevant, and it eill not be used in developments.

PNF of the fourth model is

$$P_{m4} = e^{-t(\lambda_{os}+\lambda_{ws}+\lambda_{db})} \cdot (1+\lambda_{os}t) \cdot (1+\lambda_{ws}t) . \qquad (9)$$

The fifth model takes into account diversion on WS and DBMS levels (fig. 8).

PNF of the fifth model is:

$$P_{m5} = e^{-t(\lambda_{os}+\lambda_{ws}+\lambda_{db})} \cdot (1+\lambda_{ws}t) \cdot (1+\lambda_{db}t) . \qquad (10)$$

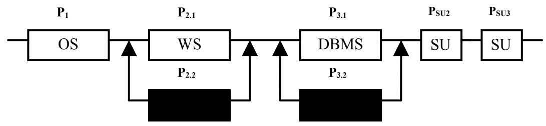

Fig. 8. Diversion on WS and DBMS levels

The sixth model takes into account diversion on OS and DBMS levels (fig. 9).

PNF of the sixth model is

$$P_{m6} = e^{-t(\lambda_{os}+\lambda_{ws}+\lambda_{db})} \cdot (1+\lambda_{os}t) \cdot (1+\lambda_{db}t) . \qquad (11)$$

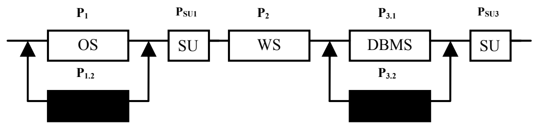

Fig. 9. Diversion on OS and DBMS levels

And, at last, the seventh model takes into account diversion on all levels of system archicture – OS, WS and DBMS (fig. 10).

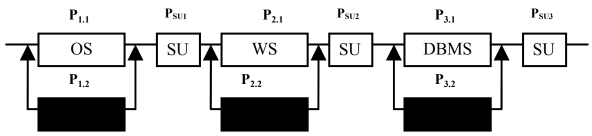

Fig. 10. Full system diversification

PNF of the seventh model is:

$$P_{m7} = e^{-t(\lambda_{os}+\lambda_{ws}+\lambda_{db})} \cdot (1+\lambda_{os}t) \cdot (1+\lambda_{ws}t) \cdot (1+\lambda_{db}t) \qquad (12)$$

In that way, using formules described above, one can figure out any diversion model PNF of given service-oriented system.

Comparative analysis of system diversion versions

We have hold dependable architecture elements analysis on OS level.

The analysis should discover, which OS through National Vulnerability Database are more reliable et the moment.

For the computation operating systems Windows XP and Mandrake Linux were chosen. Input data is a number of vulnerabilities for each OS during the period from 06.08.2004 till 26.07.2005. Thus, we have:

$$t = 8496; N_W = 38; N_L = 39.$$

Let’s consider, that as a vulnerability had appeared there was a sustem fault or failure, and so a number of vulnerabilities we can equate with a number of system failure. Then OS failure rates are:

$$\lambda_W = {N_W}/{t} \approx 0{,}00447({1}/{ч});$$

$$\lambda_L = {N_L}/{t} \approx 0{,}00459({1}/{ч}).$$

Now let's estimate diversion system element PNF on OS level, at that SU PNF, as it was mentioned earlier, isn't considered. Then, using form.4 we have:

$$P_W \approx 0,928;\quad P_L \approx 0,925.$$

It is significant that doing such kind of computations one should takes in mind a criticality of each uncovered vulnerability, and also crossed vulnerabilities of several operating systems. Therefore, it should be taken into account while reviewing results.

To compare PNF of diversion system parts with non-diversion, let's determine PNF of initial SOPhS part on OS level, using form.4. Then,

$$P_{w1} = 0{,}641;\; P_{L1} = 0{,}634\;.$$

Hence, even if PNF of Windows XP is more then Mandrake Linux has, each of system diversion models is much more reliable, than the system without diversion.

Conclusions

Therefore, developing a service-oriented pharmacological system "Medicine" we expect that it will allow to raise the prescription trustworthiness, to decrease risk of side effect progress, to get first aid in conditions of qualified doctors absence (for example, in countryside), and statistics completion of medicine implementing. Writing a prescription, a physician can withhold some features of either one or another medicine, or he can just forget some contraindications of this medicine. Perhaps, this oversight is not considerable to a doctor, but it is significant to a patient. At the best, a person, taking drugs, could feel no negative effect. At the worst, the patient can become an invalid or even die. "Medicine" was created to prevent such situation. So, we are looking forward to seeing our service in use to be calm about the health of every man, woman and child who need treatment.

References

1. Lepakhin V.K., Astahova A.V. Doctor's mistakes as a cause of drug therapy complications // Collected Works of RUCF. – M.: RUCF, 2005 – V. 12. – P. 6-10.

2. Sutcliffe A.G., Wong I.C.K. Rational prescribing for children // British Medical Journal. – 2006. – V.332. – P. 1464-1465.

3. Emelin I.V. Integration of Medicine Information Standards // Computer Technologies in Medicine. – 2004. – № 3 (96). – P. 24-29.

4. Habibullin I. Developing Web Services Using Java. – P.: BHV-Peterburg, 2003. – 388 p.

5. Anzbock R., Dustdar S. Modeling and implementing medical Web services // Data and Knowledge Engineering. – 2005. – № 55. – P. 203-236.

6. Avizienis A., Laprie J.-C. Dependable Computing: From Concepts to Application // IEEE Trans. on Computers. – 1986. – № 74 (5). – P. 629-638.

7. Kharchenko V.S. Dependability and Dependable Systems: metodology elements // Radio-electronic and Computer Systems. – 2006. – № 5 (17). – P. 7-19.

8. Gorbenko A., Kharchenko V., Popov P., Romanovsky A., Boyarchuk A. Development of Dependable Web Services out of Undependable Web Components // Collected Works of UON. – N.: UON, 2004. – V. 7. – P. 31-36.

9. Avizienis A., Laprie J.-C., Randell B., Landwehr C. Basic Concepts and Taxonomy of Dependable and Secure Computing // IEEE Trans. on Dependable and Secure Computing. – 2004. – № 1. – P. 11-33.

10. Kharchenko V.S. Dependability of Computer Systems: Problems and Results // Aerospace equipment and technology. – 2005. – № 7 (23). – P. 352-357.

Надійшла до редакції 19.01.2007

Рецензент: д-р техн. наук, проф. В.С. Харченко, Національний аерокосмічний університет ім. М.Є. Жуковського «ХАІ», Харків.