

Web-Service Selection and Quality Assurance using Anticipation

Juan Jesús Torres Carbonell

GEDES (Group on Specification, Development and Evolution of Software)

E.T.S. Ingeniería Informática. University of Granada

jesus.torres@meh.es ; jjtc28@hotmail.es

Resumen

Most Software Systems relate with others, and receive or provide web-services. As consequence, the final quality of the service depends on the interaction of several Software Systems, and the evolutionary process they undergo. A special case of service provision is the validation service: user authentication using digital certificates. We approach the evolution of Software Systems that use these services as an anticipatory process based in the previous states, the current state, and the next state, taking into account that it is an hyperincursive process in which there exist several possible future states.

1. INTRODUCCIÓN

Applications and Software Systems are not isolated systems, they have relations with the environment. Furthermore these relations can include other Software Systems. Nowadays a Software System relates with others from which it receives services or to which it provides services. So, there exists a close relation between different Software Systems to produce a concrete service. This situation implies that the final quality of the service depends on the interaction of several Software Systems that could be responsibility of one or more organizations. The way the service is provided requires interoperability and also that the Software Systems within the framework of the service provision infrastructure adapts and evolves according to the necessities.

Additionally, service provision by organizations to end users, is based, in most of the cases, in accurate user authentication. This authentication can be done, among other possibilities, through the use of digital certificates,

which seems to be the main tendency in the future. Certification Authorities are the organizations that provide information about the digital certificates' state to the Validation Authorities, and Validation Authorities are the organizations responsible for the validation of the information about user's certificate.

Because the validation of a digital certificate will be a key element in users authentication, it is prudent to use several Validation Authorities, chosen one of them in each case a validation is needed. This selection ought to be dynamic, adaptive and made according to defined strategies. If the validation activity doesn't work, all the services that need the authentication of the user won't work either. This is the main reason of the importance of the service provided by the Validation Authority.

In this case we consider the selection of the Validation Authority that provides the validation service, through web-services, as a selection that makes the decision according to the current state, the previous states, a set of specific parameters and the future state, that is to say, it is an Incursive System. Furthermore, as long as there are several Validation Authorities, we can consider that we are dealing with a Hyperincursive system with several possible states to be reached.

We use the concepts of anticipatory systems, incursion and hyperincursion, in the area of software engineering and specifically in approaching the evolution and adaptation of systems that form part of a complex system interoperable structure for the provision of services (as Service Oriented Architectures and Web Services structures). The final objective is the use of these concepts to improve the service provision quality.

We present in Section 2 the role of web-services in service provision to end users, paying

special attention to validation services provided by Validation Authorities. In Section 3 we present the characteristics of Software System explaining its anticipatory behaviour. In Section 4, we approach validation service provision as an Incurive Discrete Weak Anticipatory System. In Section 5 the role of Hyperincursion in service provision is introduced, and finally conclusion are presented.

2. WEB-SERVICES

Within organizations using web-services there can exist a base of structured information that is used by all the applications, directly or through web-services, that provide an easier access to data and facilitate reuse and development optimisation (Figure 1).

Additionally, web-services can be used by Software Systems that are part of the same organization or belong to different organizations (Figure 2). In both cases resources and data are managed and accessed using standardised interfaces that allows optimisation and improve quality.

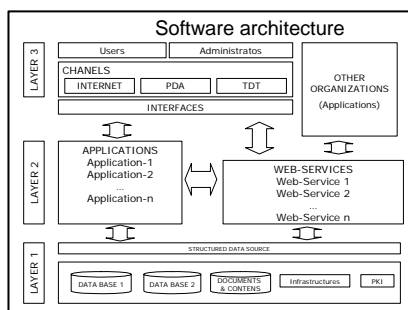


Figure 1. Software Architecture

So, we have relations between different systems. In this paper we understand system following the proposal by MDA [3]:

“A system may include anything: a program, a single computer system, some combination of parts of different systems, a federation of systems, each under separate control, people, an enterprise, a federation of enterprises ...”

“A model of a system is a description or specification of that system and its environment for certain purpose”.

These services can be provided by an inner or extern organization and produce multiple interrelations between the different elements that take part through the value chain of the provision of the service. In this chain there are interconnected dependences producing a complex relation in which there are systems or subsystems that provide services that are used by several systems. So, the final service provided to the end user could need and use web-services provided by its own organization, that are obtained through the organization Intranet, and, on the other hand, web-services provided by systems that belong to other organizations and are obtained through Internet (Figure 1).

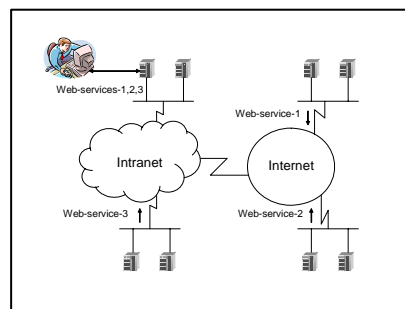


Figure 2. Service providers

The interrelation between services can be facilitated using an interchange point in which all the interested parties (users and providers) can participate. The sequence to follow in order to create, publish and use web-services is the following:

1. The organization O-1 designs and offers in the market web-services. These services are provided by this organization.
2. The organization O-1 records the service in a special directory UDDI [2] (Universal Description, Discovery and Integration), in order to be published for the applications to know how to access to it.
3. The application Ap-i needs the service and searches for its address in the UDDI directory.
4. The application Ap-i uses the web-service and sends and receives messages to and from the provider.

These interrelations of applications and systems cause that one system owes part of its

functionality to others. We face, then, the necessity to approach the evolution of the system in search of the quality, taking into account the web-services it uses.

2.1. Digital Certificates, PKI and Validation

Users must authenticate when using applications. As pointed out in the introduction, one way of identifying users is the use of digital certificates, that are documents provided by Certification Authorities. Certification Authorities are entities specialized in digital certificates management (provides, revocates). A digital certificate is a document with information about the issuer, the Certification Authority and the certificate itself: Version, Signature, Validity, Key Usage Authority Key Identifier, etc. The certificate can be used with different objectives: electronic signature, ciphering process, authentication, non repudiation, securing Internet communications, etc.

A PKI (Public-Key Infrastructure) is an infrastructure that offers services implemented and provided using public-key concepts and techniques. So, a PKI is based in a Trusted Third Party (TTP) concept that try to answer the question: How can I trust in a certificate?. The validity of a certificate is the trust in the fact that the public key that it contains belongs to the user indicated as such in the certificate. That is to say, a PKI is a set of protocols, services and standards that allows the existences of applications and services based in public key cryptography. In the environment of a PKI a certification is the act of binding a name with a public-key. This relation is a signed structure: public key certificate, and the Certification Authority is responsible for its accuracy.

Additionally to the PKI and the services that it provides, we can identify the role of the Validation Authority and the validation service. A Validation Authority is a structure or organization responsible of checking the validity of a certificate. This certificate can be provided by any Certification Authority. So the Validation Authority provides confidence in the certificate and the Certification Authority.

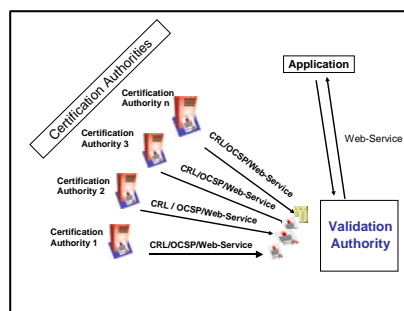
The validation process can be done by the Software Systems that need the service by themselves. Nevertheless the self-provision of the

service requires a continuous effort to include new Certification Authorities (Figure 2) or new kind of certificates, together with the necessity for the service to be provided 24x7 what requires more economical, technical and human resources. So the usual solution is to use the web-services provided by a different organization that specialize in the provision of these services.

As a number of Certification Authorities can exist, not all the certificates have the same structure. So, it is very useful to use the services provided by the Validation Authority that is the responsible for adapting its interfaces to these different structures.

The adaptation process to these services provided by Validation Authorities followed by Software Systems, is not known a priori, and there exists an anticipatory behaviour as we explain in the following section.

Figura 3. Validation service



3. SOFTWARE SYSTEMS AND ANTICIPATION

We have previously presented our point of view of Software Systems evolution under the perspective of Anticipation [4], [5]. Following [1], we have presented the way we consider the definition of an Incurive Discrete Strong Anticipatory System: an incurive discrete system is a system which computes its current state at time t , as a function of its states at past times, ..., $t-3$, $t-2$, $t-1$, present time, t , and even its states at future times $t+1$, $t+2$, $t+3$, ..., that is to say:

$$x(t+1)=A(\dots,x(t-2),x8t-1),x8t),x(t+1);p) \quad (1)$$

In the framework of Software System, we can similarly write:

$$S_{t+1}=M(\dots,S_{t-2},S_{t-1},S_t,S_{t+1};OS) \quad (2)$$

Where:

- ... , St-i, Are the previous states of the Software System known through the Memory of the system.
- St Is the current state of the Software System.
- St+1 Is the next state of the Software System, the state at the following evolutionary instant.
- OS Is the parameter that indicates the evolutionary modifications to be performed.

That is to say, the next state of the evolutionary process of the Software System is decided according to the previous states, the current state and also the following state itself, applying additionally and evolutionary operator that is defined by the parameter OS.

In the anticipatory framework the coexistence of strong and weak anticipation is possible when considering the evolution of Software System. The strong or weak character will depend on the kind of knowledge about the state t+1 and following ones. So, if the knowledge is the result of a model of the system, then it is said to be weak anticipation.

We proposed that anticipation between subsystems could be considered as weak anticipation [6]:

"...That is to say, subsystems could manage a model of other subsystems and anticipate the future states according to that model, as a complement to strong anticipation, characteristics of the intra-anticipation, in which a system/subsystem anticipates its own evolutionary process..."

In this case we are dealing with a case of Software System's inter-anticipation [6] in which the system that undergoes evolutionary processes uses a model of the system (or systems) that provides services through the web-services. Thus, we have:

$$\sum_{j=1}^n \sum_{k=1}^m Model(S_j WS_k) \quad (3)$$

Where $S_j WS_k$ is the model that the system that receives the service has of the web-service WS_k provided by the system S_j . The model is the formal representation of the functional

characteristics that must be taken into account during the evolutionary process.

Additionally, we have three important elements: the structure of the web-service, the SLA (Service Level Agreement between provider and user) and the indicators that should be established. So, the system that receives the web-service provided by System-j, particularizes the model (3) as a function of these three elements:

$$FunctionWebService_{jk} = \{WS_{jk}, SLA_{jk}, indicadores_{jk}\} \quad (4)$$

Where:

1. WS_{jk} : Is the web-service's structure, and it can be represented as a result of: detected errors, values out of rank, etc.
2. SLA_{jk} : that could be represented as: response time, percentage of errors, availability, penalizations, etc.
3. $Indicadores_{jk}$: established according to the services provided and the SLA.

As consequence, we have that each evolutionary modification (OS_i) is the necessary change to obtain:

$$Model(S_j WS_k)_t = [FunctionWebService_{jk}(WS_{jk}, SLA_{jk}, indicadores_{jk})] \quad (5)$$

That is to say, depending on the value of the function $FunctionWebService_k(WS_k, SLA_k, indicadores_k)$ in time t (the result of the provision of the web-service in t), that is the model of the system in relation with the web-service in instant t, an operator OS_i will be selected and will be applied to produce an evolutionary process in the system. Furthermore, there can be several web-services provided by one or several systems. So, we have that the set of necessary modifications, OS_{global} , will be:

$$\sum_{j=1}^n \sum_{k=1}^m Model(S_j WS_k)_t = \quad (6)$$

$$\sum_{k=1}^{j=n} [FunctionWebService_{jk}(WS_{jk}, SLA_{jk}, indicadores_{jk})]$$

Depending on the concrete model in time t, and the knowledge of the evolution followed through previous states, we have an expected

service's quality, and an evolutionary operator can be applied:

Expected Quality	$Model(S_j WS_k)_t$ $= FunctionWebService_{jk}(WS_{jk}, SLA_{jk}, indicators_{jk})_t$			
	Range 1	Range 2	...	Range n
Range 1	OS ₁	OS ₂	...	OS _n
Range 2	OS _{n+1}	OS _{n+2}	...	OS _{2n}
...

Tabla 1. Version Sapce

In Table 1 the set of evolutionary operators that can produce modifications in the Software System is established, that is to say, the set of operators that produce the Version Space of Possible Configurations. This Version Space will include the possible and viable configurations that obey to the temporal, ontological and functional necessities [7].

There are several choices. As consequence of the information available an operator can be applied that replaces the request of a web-service by a different one. For example if the quality of the service provided by WS_{il} , goes down it can be replaced by the web-service WS_{jk} by the modification of the request. Now we have all the necessary elements for the system (or the Administrator) to make a decision about the next state in the evolutionary process: Previous states (... , St-2, St-1), Current State (St), $FunctionWebService_{jk}$, Expected quality. The only remaining step is to decide which are the "weights" of the relations between these elements in order to decide the evolutionary modification (to reach the next state), that is represented by OS (Figure 3).

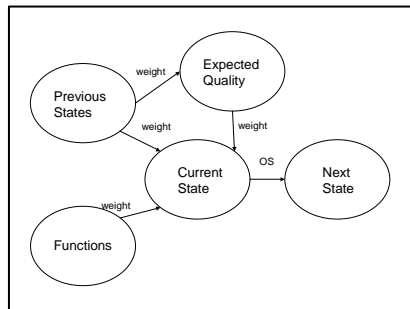


Figura 4. Reaching next state.

4. VALIDATION SERVICE

In the case of the selection of the validation web-service we have:

$$ValidationService-WebService_{jk} = \{WS_{jk}, SLA_{jk}, indicadotors_{jk}\} \quad (7)$$

Where:

WS = Is the reply that the Validation Authority sends. This reply consists in an OSCP reply or a full web-service including certificate data extraction, depending on the agreement between the requester and the provider of the service.

SLA = Is the Service Level Agreement, that is to say, the obligation of the service provider concerning the quality of a set of elements of the service. In this example, we can identify the following elements in the SLA:

1. Time of response = TR. This item is the obligation about the expected time and maximum delay in answering a request of service of the Validation Authority. We can consider 3 grades:

$$\begin{aligned} TR &= < t1 \\ t1 < TR &= < t2 \\ t2 < TR & \end{aligned}$$

2. Denial of services = DS. This item represents the number of occasions in which a request of service is not provided by the Validation Authority. We can consider 3 grades:

$$\begin{aligned} DS &= < lim1 \\ lim1 < DS &= < lim2 \\ lim2 < DS & \end{aligned}$$

3. Time without service = TWS. This item represents the maximum time without service provision that is acceptable. We can consider 3 grades:

$$\begin{aligned} TWS &= < t1 \\ t1 < TWS &= < t2 \\ t2 < TWS & \end{aligned}$$

4. Delay in including CAs = DI. This item represents the maximum delay in accepting a new certificate since it is in the market, or a new Certification Authority by the Validation Authority, since the moment in which the CA provides services. We can consider 3 grades:

$$DI = < t1$$

$$t1 < DI \leq t2$$

$$t2 < DI$$

5. Programmed interruptions = PI. This item represents the number or interruptions of the services programmed by the Validation Authority. The values “pi-x” could be weighted according to the duration of the interruption. We can consider 3 grades:

$$PI = < pi1$$

$$pi1 < PI \leq pi2$$

$$pi2 < PI$$

Indicators = We consider the following indicators:

1. User satisfaction. This indicator could take a value in the interval [0, ... 5], according to the criteria of the Administrator of the system that requests the services of the Validation Authority.

2. Application Modifications = AM. This indicator represents the number of modifications that the system that requests the services of the Validation Authority must suffer. We can consider 3 grades:

$$AM = < am1$$

$$am1 < AM \leq am2$$

$$am2 < AM$$

3. Available bandwidth used in the communication with the Validation Authority

With the information obtained from *WS*, *SLA* and *Indicators*, we have a value of the function $ValidationService-WebService_{jk} = \{WS_{jk}, SLA_{jk}, indicators_{jk}\}$. Additionally to this value we must establish an Expected Quality of the service provided:

Expected Quality: This must be a simple index that reflects the quality. One possibility is to include a series of items, for example those considered in the function $ValidationService-WebService_{jk}$: So, suitable intervals can be established:

- Time of response = TR $\subset [t_0, t_1]$
- Denial of services = DS $\subset [ds_0, ds_1]$
- Time without service = TWS $\subset [tws_0, tws_1]$
- Delay in including CAs = DI $\subset [di_0, di_1]$
- Programmed interruptions = PI $\subset [pi_0, pi_1]$

Finally, evolutionary modification (OS) to perform could include:

1. Request service to Validation Authority i.
2. Cancel the service request to Validation Authority i.
3. Request service to other Validation Authority when TR > t1
4. Fire an alert when TR > t2. This case could imply the intervention of the System's Administrator.
5. Request service to other Validation Authority when DS.
6. Fire an alert when DS. This case could imply the intervention of the System's Administrator.
7. Shift from Validation Authority 1 to Validation Authority 2.
8. etc.

Obviously, the set of OS listed above is a very simple example. A complex reasoning system is required to take into account all the items formerly presented and represented in Figure 4. This is future work to be done.

5. THE ROLE OF HYPERINCURSION IN SERVICE PROVISION

A Hyperincursive Discrete Anticipatory System is an Incurive Discrete Anticipatory System that produces several iterations at each time step. Dubois exposes [1]:

“Karl Pribarn asked me (by email, after CASYS '99 conference):

How can an anticipatory hyperincursive system be modeled without a future defined goal?.

A Hyperincursive Anticipatory System generates multiple potential states at each time step and corresponds to one-to-many relations. A selection parameter must be defined to select a particular state amongst these multiple potential states. These multiple potential states collapse to one state (amongst these states) which becomes the actual state”.

Additionally to the Incurive Discrete Weak Anticipatory System character that a Software System that uses web-services provided by other systems has, we want to approach briefly in this paper its Hyperincursive Discrete Anticipatory System character as well. As we have previously stated Software System evolution is an HDSA process [5]:

1. The input available to make an evolutionary decision is composed by the information about previous states, current state, function $ValidationService-WebService_{jk}$, and also the expected quality.
2. From the Software System current state, S_t , a set of Potential States is taken into account in deciding the following instant of the evolutionary process, according to the set of selection parameters derived from the knowledge obtained from function-service-
 1. Potential states may include:
 - To send a new request of service provision to the same service provider.
 - To send a request of service to a different service provider.
 - To wait some more time for the response.
 - To fire an alert.
 - To send a message to the system's Administrator.
 - Etc.

The concrete evolutionary action that will be applied, is not known *a priori*. Then, multiple potential adaptations can be produced.
3. Through the selection of the operator to be applied (OS) the following evolutionary state is reached, it will be the next state S_{t+1} . This implies that the system behaves as a Hyperincursive Discrete Anticipatory System, because all the possible evolutionary actions particularize into one action that will produce the selected next state.

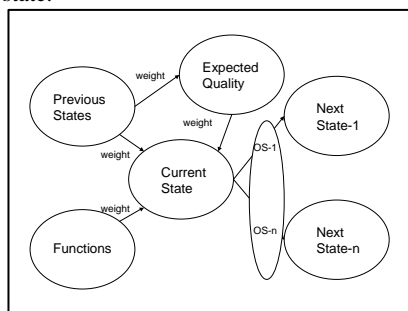


Figura 5. Hyperincursion in OS application.

So, service selection is a Hyperincursive process that is followed by Software Systems that

use services provided by other systems and make evolutionary decision during their relations in order to guarantee the highest level of quality as possible.

6. CONCLUSION

We have presented the importance of the service provision in order to provide complex services to final users. The trustworthiness and availability of web-services are crucial because if a web-service fails the service to the end user can't be provided (for example, the identity of the user can't be checked). Software Systems that rely in the web-service provision must evolve and adapt according to the conditions and quality of the services received. From our point of view, Software Systems behave as Incursive Discrete Weak Anticipatory Systems in the process of service selection, based in the availability of a model of the services provided, the previous states of the system, the current state and the expected quality. Moreover, a Software System during the process of service selection behaves as an Hyperincursive Discrete Anticipatory System, because there exist several potential evolutionary actions and only one is taken to produce the next state. Our future work will include a very important item not yet approached: the reasoning process to make the decision using the weights of the relations between the different items that are involved.

Referencias

- [1] Dubois, D.M. (2000): "Review of Incursive, Hyperincursive and Anticipatory Systems- Foundation of Anticipation in Electromagnetism" in Computing Anticipatory Systems: CASYS'99 - Third International Conference, edited by D.M. Dubois, AIP Conference Proceedings 517, Melville, New York, 2000, pp. 3-30.
- [2] OASIS 2004. UDDI UDDI Version 3.0.2, UDDI Spec Technical Committee Draft. <http://uddi.org/pubs/uddi-v3.0.2-20041019.pdf>
- [3] OMG 2003. "MDA Guide Version 1.0.1" <http://www.omg.org/docs/omg/03-06-01.pdf>
- [4] Torres-Carbonell, J.J. and J. Parets-Llorca (2001): "Software Evolution. What kind of Evolution?", in Computing Anticipatory

Systems: CASYS 2000 – Fourth International Conference, edited by D.M. Dubois, AIP Conference Proceedings 573, Melville, New York, 2001, pp. 412-421.

- [5] Torres-Carbonell, J.J.; Parets-Llorca, J.; Dubois, D.M. (2002). “Software Systems Evolution, Free Will and Hyperincursivity”. International Journal of Computing Anticipatory Systems, CHAOS, Liège, Belgium. Volume 12. Soft Computing and Computational Intelligence; Cognitive, Neural, and Psychoanalytical Anticipation; Computer Science, Data and Knowledge Systems. Partial Proceedings of the Fifth International Conference CASYS'01 on Computing Anticipatory Systems, Liège, Belgium, August 13-18, 2001, D. M. Dubois (Ed.), pp. 3-24.
- [6] Torres-Carbonell, J. J.; Paderewski-Rodríguez, P. and J. Parets-Llorca, (2004). “Software Systems Intra/Inter-Anticipation”, in Dubois D. (Ed.): “Computing Anticipatory Systems: CASYS 2003 – Sixth International Conference”. Published by The American Institute of Physics, AIP Conference Proceedings.
- [7] Torres-Carbonell, J. J.; Parets-Llorca, J. and D.M. Dubois (2005) “Emergencies Scope of Complex Systems. Software Version Space and Incursivity” , published in the book ANTICIPATIVE AND PREDICTIVE MODELS IN SYSTEMS SCIENCE, Volume I, pp 45-50; by George E. Lasker and Daniel M. Dubois (Eds), published by the IAS, in Windsor, Ontario, Canada in May 2005. ISBN 1-894613-49-X