

Software Confederations and Alliances

Jaroslav Král and Michal Žemlička

Department of Software Engineering, Faculty of Mathematics and Physics,
Charles University, Prague, Malostranské nám. 25, 118 00 Praha 1, CZ*
kral,zemlicka@ksi.mff.cuni.cz

Abstract. Global processes imply that the information systems have the architecture of virtual peer-to-peer networks of autonomous permanent services. It is typical for IS supporting *e*-government. We show that there are two variants of such networks: Confederations with quite stable relations (e.g. between the IS of the offices) and alliances with temporal links (typical for *e*-commerce). Alliances must apply message format standards like SOAP whereas confederations can (and often should) use domain-oriented message formats.

1 Introduction

There is a growing need to interconnect (integrate) almost independent or autonomous large software components (legacy information systems, third party products, newly developed autonomous software units) working as permanently available services (such systems are called *software confederations* – SWC). The corresponding software systems usually must have a structure of a peer-to-peer network of large components (i.e. the system has a coarse granularity) communicating asynchronously via messages of appropriate structure/format.

An important feature of SWC's is how loosely are its constituent components related – see the following list:

1. IS of an international enterprise having the structure of a network of autonomous organizational units (divisions). IS of the enterprise is formed by a peer-to-peer network of the IS's of the divisions and by some additional units serving the whole system of the enterprise. This architecture simplifies the integration of new divisions and/or enterprises as well as the selling out some divisions or splitting the enterprise into smaller ones.
2. Information system of *e*-government ([KŽ01]). Such information systems must be built as a network of the information systems of particular offices¹.
3. The collaboration between the information system(s) of the given enterprise and the information systems of suppliers as an implementation of supply chain management (SCM) and customer relationship management (CRM).
4. The IS supporting a purchase coalition of car vendors. The aim of the coalition is to better the position of the coalition members with respect to the suppliers of the automobile parts and to cumulate the member orders.

* Supported by the grant of the Grant Agency of Czech Republic number 201/02/1456.

¹ Another solutions are not feasible ([KŽ03,KŽ01,Ro02]).

5. A highly unformal open association of health organizations (physicians, hospitals, laboratories, database services, etc.) forming an information system intended to simplify, enhance and speed up health care.
6. Open dynamically changing network of cooperating software systems necessary for the running of e -commerce.

2 Confederation Types and Message Formats

In all the above cases the system is a peer-to-peer network of a highly autonomous components used as black boxes (i.e. their interface only is known).

The cases 1 and 2 describe the situation when the network of (closely related) autonomous components is comparatively small and not too dynamic. It results into the situation when a quite complex message formats are required and can be agreed. The format of the messages tends to follow knowledge domains of the enterprise staff and to the requirements of the enterprise activities. The formats can be agreed in small groups. It leads to complex and 'declarative' ad-hoc data formats. We shall call such systems *confederations*.

In the cases 3 and 5 any member of the network must look for a partner (addressee of a message) in principle again for any new message. In the case of e -commerce the topic and the intuition related to it is world-wide known. The format of the messages can be standardized. An example is SOAP [W3Ca]. We call such systems *alliances*. The problem with SOAP and similar standards is that it vary too quickly and that it is too implementation oriented and too complex.

The case 4 is close to the confederations. The crucial/kernel notions and/or constructs of the confederative attitude are: (i) The system is formed as a changeable collection of permanently active cooperating autonomous services – autonomous components; the system therefore has some common features with the services in human society; (ii) the system use an adaptive middleware; (iii) the autonomous components are nodes of a (virtual) peer-to-peer network.

Confederation philosophy is more suitable for the integration of applications than object orientation – especially in the case of larger or real-time applications.

3 Confederations

A straightforward requirement of any user of the information system of e -government is that the user need not bother about the internal structure of the system. To achieve it the confederation must have its own user interface different from the interfaces of its components (note that the components can be complex information systems with their own interfaces). The interface should be as flexible as possible. A good solution of this problem is the implementation of the user interface (UI) as a two tier structure consisting of: proper client software (often a browser), and a software unit UI transforming the user message formats (usually HTML) into the internal message formats of the confederations (usually XML formats). The purpose of UI is:

1. enabling the communication of any user with the system in the message formats reflecting the knowledge domain of the user and the requirements/restrictions of his browser,
2. to hide internal structure of the confederation and to create the impression that the system is one integrated unit,
3. to hide the structure of the internal messages of the system (as the internal messages can be implementation dependent),
4. to offer different interfaces for different user groups.

The best way to implement UI is probably the use XSLT applications and to design UI as an autonomous component (a new peer of the peer-to-peer network see e.g. [KŽ02, LN02]).

The quality of confederations and alliances depends substantially on the properties of component interfaces, especially on their stability. The software implementation techniques, implementation details and even implementation philosophy vary too quickly. The message formats should be user oriented, i.e. declarative rather than than implementation-oriented (i.e. procedural, respective object-oriented).

The declarative property means that the message formats should reflect the knowledge domain behind the requirements on the functions of the particular component (e.g. the knowledge domain of bookkeepers). The declarative interface can depend on the needs of partners (the components communication with the given one, the user interface components inclusive).

As the declarative interface is quite far from implementation, it can happen that it could not offer all the possibilities of a procedural interface using implementation details knowledge. This problem can be solved if we design the gate as a two-tier structure. The basic tier (*basic gate* or simply *gate*, G) provides procedural implementation interface offering the access to all features of the corresponding component, the second tier (*front-end gate*, FEG) transforms the declarative messages into sequences of procedural-oriented messages for G and vice versa. The problem is similar to the problem of the implementation of two-tier user interface. In fact we can use the same techniques as for user interface components. Especially it is desirable to implement FEG as an autonomous component. FEG is similar to a wrapper. The requirement, however, that it is an autonomous component is from the engineering point of view crucial.

4 Alliances (Semiconfederations)

Alliances formed temporarily during e.g. *e-trade* processes are too dynamic to have their own user interface. Their autonomous components (and their users) must be aware about the ‘internal’ structure – i.e. they must know who are the other members of the coalitions – the business partners. The partners are not known before the first step of the *e-commerce* process is ready (compare WSDL [W3Cb] and UDDI). The properties of the partners (e.g. trustworthiness) must be evaluated. Such a system has almost no stable communication links.

The format of messages in *e-commerce* must be standardized as the communication is too dynamic to allow any ad-hoc definition of message formats

and/or communication protocols by communicating bodies. But experience indicates ([KŽ,IBM]) that the standards are not stable (fixed) enough yet. This problem can be solved by using wrappers/FEG's.

5 Conclusions

Confederative attitude is a powerful tool. Recommended software engineering methods and/or properties like prototyping, incremental development, modifiability, openness, testability, etc. are very simple to implement and use in the confederative environment. It is not difficult to integrate legacy systems or third party products as well as newly developed autonomous components.

Confederative philosophy enables software to gain properties similar to the properties of other high-tech products. Confederative philosophy is now a matter of everyday practice giving the firms using it a competitive advantage and independency of software vendors. This may be the reason why the most of the software vendors does not propagate it.

The ways to implement an information system were completely unimportant for central executive officers (CEO) as well as for middle level managers. It is not true in the case of confederations and alliances. Confederative attitude is a new challenge and/or opportunity for managers.

References

- [IBM] IBM: *BPE4WS – Business Process Execution Language*.
<http://www.ibm.com/developerworks/library/ws-lpl>.
- [KŽ01] J. Král and M. Žemlička. Electronic government and software confederations. In A. M. Tjoa and R. R. Wagner, editors, *Twelfth International Workshop on Database and Experts System Application*, pages 125–130, Los Alamitos, CA, USA, 2001. IEEE Computer Society.
- [KŽ02] J. Král and M. Žemlička. Autonomous components. In M. H. Hamza, editor, *Applied Informatics*, pages 125–130, Anaheim, 2002. ACTA Press.
- [KŽ03] J. Král and M. Žemlička. Software Confederations – An Architecture for Global Systems and Global Management. In: S. Kamel: *Managing Globally with Information Technology*. IRM Press, Hershey, PA, USA.
- [KŽ] J. Král and M. Žemlička. *Software Confederations and Manufacturing*. To appear.
- [LN02] W. Löwe and M. Noga. A lightweight XML-based middleware architecture. In M. H. Hamza, editor, *Applied Informatics*, pages 125–130, Anaheim, 2002. ACTA Press.
- [Ro02] D. Rowe. e-government motives & organisational framework. In J. Pour and J. Voříšek, editors, *Systems Integration 2002, Conference Presentations*, pages 93–99, Prague, Czech Republic, 2002. Prague University of Economics.
- [W3Ca] W3 Consortium. Simple object access protocol. A proposal of W3C consortium. <http://www.w3.org/TR/SOAP>.
- [W3Cb] W3 Consortium. Web service definition language. A proposal of W3 consortium. <http://www.w3.org/TR/wsdl>.