Title: Coupling component systems towards systems of systems

Authors: Frédéric Autran[1], Denise Cattan[2], Jean-Luc Garnier[3], Dominique Luzeaux[4], Marc Peyrichon[5], Jean-René Ruault[6]

[1] Senior Expert, System Design Center, EADS Defence & Security, frederic.autran@eads.com
[2] Scientific Advisor, SPIRULA, denise.cattan@wanadoo.fr
[3] Architecture Framework Technical Coordinator, THALES, jean-luc.garnier@fr.thalesgroup.com
[4] Director of the Technical Center for Information Systems, DGA, dominique.luzeaux@polytechnique.org
[5] SoS Architect, DCNS, marc.peyrichon@dcn.fr
[6] Naval Battlelab Architect, Service d'Architecture Inter-Système, DGA, jean-rene.ruault@dga.defense.gouv.fr

[123456] All authors are members of the Task Group "Systems of Systems" of the AFIS (Association Française d'Ingénierie Système)

Abstract
Systems of systems (SoS) are a hot topic in our "fully connected global world". Our aim is not to provide another definition of what SoS are, but rather to focus on the adequacy of reusing standard system architecting techniques within this approach in order to improve performance, fault detection and safety issues in large-scale coupled systems that definitely qualify as SoS, whatever the definition is. A key issue will be to secure the availability of the services provided by the SoS despite the evolution of the various systems composing the SoS. We will also tackle contracting issues and responsibility transfers, as they should be addressed to ensure the expected behavior of the SoS whilst the various independently contracted systems evolve asynchronously.

Introduction
Systems of systems (SoS) can be defined loosely as a combination of systems in order to fulfill some kind of capability, with the additional fact that the composing systems should have operational and managerial independence. We will not delve into the current debate of looking for the appropriate definition, since our aim is to start from a real-world generic example and address concrete issues, which can be used later to feed the current debate.
Henceforth we will deal with several systems that already provide services to their customers/users, and that are coupled with some new structure – which we dare to call a SoS – that provides new (emergent) services to the customers/users. The coupling creates added value on the one hand, as new services are available, but it increases the appearance of failure modes within the whole chain value on the other hand.
We will show that a straightforward extension of the standard functional dependence coupling matrix can be used to provide adequate answers.

Definitions: coupling matrix and system of systems
A key driver to understanding the non-triviality of the current debate on SoS is that, following the generally accepted definition, a system delivers products and/or services. Hence the combination of systems gives birth to a tangle of products and services, which justifies the search for an encompassing concept but adds to the general confusion. The merging of tangible and immaterial value creating entities actually contributes to the complexity of the resulting structure.

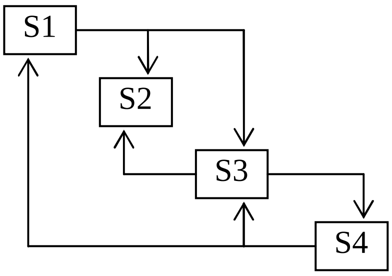
Among the popular definitions of SoS, Mark Maier's definition [MAI98b] underlines the following properties:

- *Operational independence of the elements: if the SoS is disassembled into its component systems the component systems must be able to operate independently in an efficient way. The SoS is composed of systems which are independent and useful in their own right.*
- *Managerial independence of the elements: the component systems not only can operate independently, they do operate independently. The component systems are acquired and integrated separately but maintain a continuing operational existence independent of the SoS.*
- *Evolutionary development: the SoS does not appear fully formed. Its development and existence is evolutionary with functions and purposes added, removed, and modified with experience.*
- *Emergent behavior: the system performs functions and carries out purposes that do not reside in any component system. These behaviors are emergent properties of the entire SoS and cannot be claimed by any component system. The principal purposes of the SoS are fulfilled by these behaviors.*
- *Geographic distribution: the geographic extent of the component systems is large. Large is a nebulous and relative concept as communication capabilities increase, but at a minimum it means that the components can readily exchange only information and not substantial quantities of mass or energy.*

Our approach to managing the creation of value obtained through the combination of systems is to adopt a service-oriented picture, and adapt current "product-driven" system engineering tools and use them as "service-driven" SoS engineering tools. Indeed when combining systems for which products are exchanged, consumed, or transformed, the resulting added value is a priori not greater than the sum of all added values of the component systems. However when considering services (which are immaterial, can be composed, and have added-value for the consumer and the provider in a predefined context of use – cf. ISO/CEI20000) the story changes. Collaboration between service-providing systems allows realizing higher-level services which contribute to the added value of the target SoS. Henceforth in the sequel, we will mainly discuss services and relegate products to the background.

The engineering tool used extensively in this paper is the N² dependence coupling matrix [MEI98, MEI02]. It is used to combine the components into sub-systems, including the communication means. One aims at obtaining sub-systems with a strong/high internal cohesion and a loose external coupling.
In our context of SoS, the components are systems. The connections/links/interfaces vehicle combinations of products and services. Such combinations can be either sequential or more complex (parallel, braiding…) combinations. By identifying dependence and collaboration between these service-providing systems, one wants to enhance and preserve the added value of the target SoS and to manage the configuration of the latter.

Figure 1 illustrates the former notions with an example applied to functional flows. On the left side the coupling between four systems (S1, S2, S3 and S4) is represented by a flow diagram. The right side shows the corresponding coupling matrix.



S1	Flow S1→ S2	Flow S1 → S3	
	S2		
	Flow S3 → S2	S3	Flow S3 → S4
Flow S4 → S1		Flow S4 → S3	S4

Figure 1: Coupled systems and the resulting coupling matrix.

The systems building up the SoS lie on the diagonal of the coupling matrix, whereas the flow exchanged between a source component *i* and a target component *j* of the SoS lies on the corresponding *(i,j)* cell (the row corresponds to the source component while the column is the target component).
As an additional feature, we can associate to each cell various information, such as the necessary resources in order to fulfill the relevant service, or critical parameters/constraints to take into account for the safety or the nominal functioning of the systems. Actually this additional information can be organized so as to yield various architecture views of the SoS, similar to what is commonly done by system architects.
Furthermore, we will detail how it is possible to use this matrix to depict dependencies other than flows such as physical interfaces, contractual management or legal rules.

One of the advantages of this representation is to yield an easy way to read emerging functions and services: if there is a path – or a set of paths – leading from a source cell *(s,s)* to a target cell *(t,t)*, i.e. a chain of dependencies $[(s,s),(s,i_1),(i_1,i_2),\ldots,(i_n,t),(t,t)]$, then the combination (sequential, parallel, etc.) of all that services defines a new service, that can be denoted as emerging since it was not foreseen initially. It can be defined informally in natural language, as can be seen from the example below, but more interestingly it can be defined formally when one looks at the associated resources and when one knows how to compose formally the services.
We will not detail any formal technique in this paper that can be used to define compositionally the emerging services, but they are similar to what is used in process calculus (e.g. Milner's pi-calculus [MIL99]) in mobile communication theory. Furthermore, formal verification techniques can be defined that rely on particular logics, like linear logic (cf. Girard's and Lafont's work on linear logic proof nets [Abrusci 95]), that take resource consumption into account. This shows how the seemingly trivial representation above can be used extensively throughout the design and verification processes within the engineering of the SoS.

Context and case study
We will use below a case study to explain these concepts and to elaborate the functional dependence coupling matrix. This case study is based upon a NCOIC case study, extracted from the Telecommunication Industry Association (TIA) TR88 scenario, and is the EMS SAFECOM scenario.

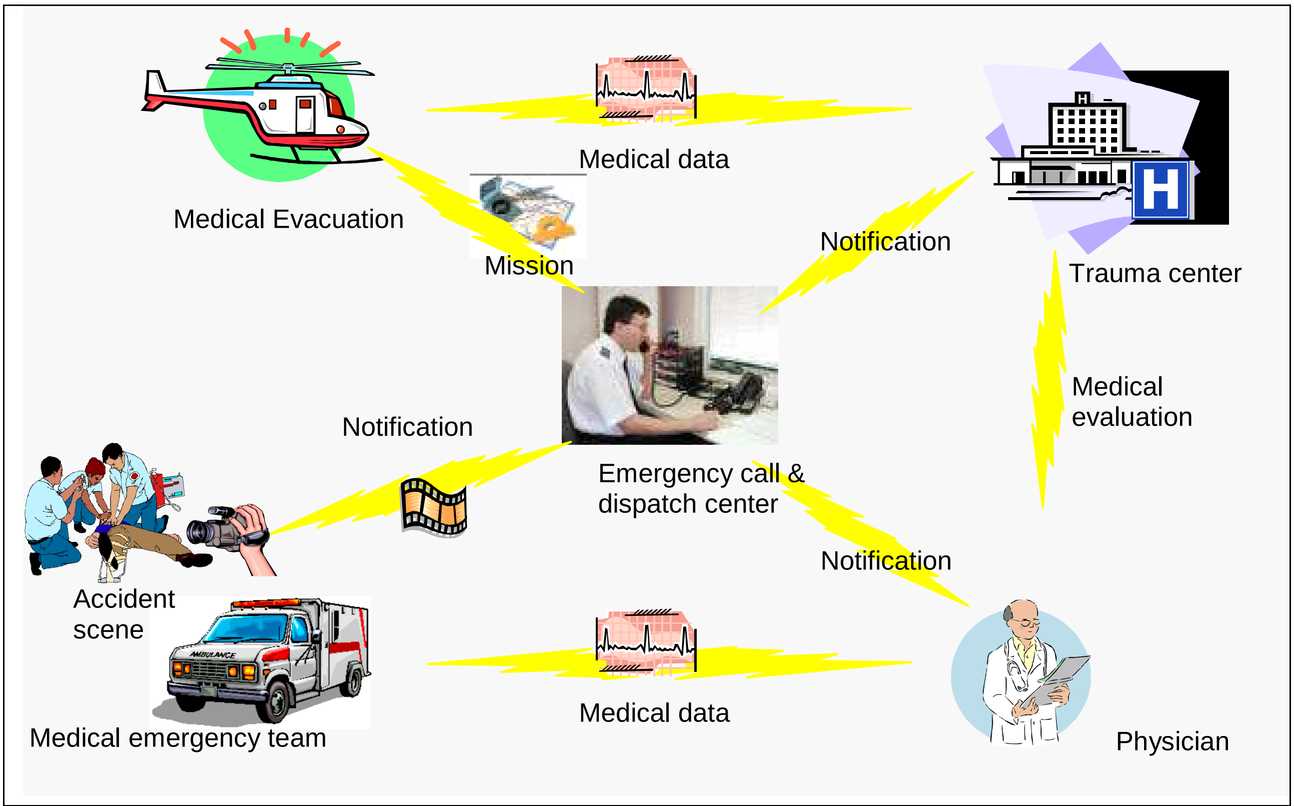


Figure 2: Operational view of the emergency medical system (EMS).

The scenario deals with an accident that involves a patient who has to be evacuated to a medical rescue center. Various medical emergency means can be used, depending on the initial medical evaluation and the evolving condition of the patient. The medical rescue center depends also on these data. Figure 2 illustrates the case study.

Integration of systems provides new capabilities to the whole SoS. These new capabilities, which lead to improved global performance, are:

- Optimization of medical and emergency resources,
- Reduction of overall processing time, resulting in reduction of accident death rate.

These emergent properties and new capabilities allow to:

- Assess medical care (telemedicine, tele-diagnosis),
- Send relevant and optimized resources to the accident,
- Assign patients to the relevant available hospital, depending on its care needs and of the availability of hospitals' services,
- Fast patient evacuation to the identified hospital,
- Prepare hospital's resources function of the patient diagnosis.

These new capabilities and emergent properties rely on the integration of systems into a SoS and the organized composition of the following services:

- The automatic call notification (ACN) provides:
  - A notification of the accident to the emergency call center.
- The emergency call center provides:
  - Localization of the accident to the emergency medical team,
  - Advised route communication to the emergency medical team,
  - Accident notification to the physician,

- Evacuation mission communication to the helicopter,
  - Plan transportation to the helicopter,
  - Patient evacuation state notification to the hospital.
- The emergency medical team (EMT), with an ambulance, provides:
  - Accident scene reconnaissance to emergency call center,
  - First medical assessment to the emergency call center,
  - Exams results and medical data (hull scan, electrocardiogram…) to the physician,
  - Patient evacuation state notification to the hospital.
- The physician provides:
  - Exams consigns to the emergency medical team,
  - Medical diagnosis to the hospital.
- The helicopter provides:
  - Patient vital data to the hospital,
  - Patient evacuation state notification to the hospital.
- The hospital provides:
  - Patient evacuation state notification to the emergency call center.

Resulting coupling matrix

The previous description is easily translated into the following coupling matrix.
This matrix allows verifying loops of interactions between systems (e.g. servo control) and the source and pit of information.

Automatic Call Notification	• Accident Notification				
	Emergency call center	• Accident localization • Advised route communication	• Accident notification	• Evacuation mission communication • Plan transportation	• Patient evacuation state notification
	• Accident scene reconnaissance • Medical assessment	**Medical emergency team / Ambulance**	• Exams results • Medical data (EEG, ECG…)		• Patient evacuation notification
		• Exams consigns	**Physician**		• Medical diagnosis
				Helicopter	• Patient evacuation state notification • Patient vital data communication
	• Patient evacuation state notification				**Hospital**

Table 1: Coupling matrix for the EMS scenario.

Dependence matrix: compatibility and interoperability issues

The previous discussion has eluded the environment, which plays an important role in an SoS, as the latter is very often an open system. However engineering does not cope too well with openness, especially when safety and configuration management are important issues and have a direct impact on the global ownership cost. A solution is to internalize the environment within the SoS, i.e. to model its key features and consider it as an additional system which is interfaced with the other component systems. This is all the more relevant of the various behaviors exhibited by the original SoS which are robust to a large class of disturbances, as the assumed model of the environment is obviously a simplified representation. From now on, we assume this step has been performed: in our case study, this implies the introduction of an additional system, denoted by the "environment", which has a strong coupling with the "medical emergency team/ambulance" (modeling traffic and weather conditions), a weak coupling with the "helicopter" (weather conditions), and a functional coupling with the "emergency call center", which corresponds to the service exchanged in order to control the various disturbances (weather, traffic, etc.).

Since the SoS consists of several interconnected systems which have been designed a priori without knowledge of each other, the various assumptions about the external world of each system may conflict, leading to compatibility problems (e.g. electromagnetic compatibility between the medical devices on board the ambulance and the transmitter used to communicate with the emergency call center). These are a special case of interoperability issues, which are crucial to allow any service exchange between technical systems. Interoperability is not restricted to the existence of physical links between the systems. It occurs at various levels; for instance, NATO defines three levels of interoperability for military systems:

- Physical interoperability: a communication link must exist. This link can be wired or wireless, and is not necessarily IT-based, e.g. voice can be used as a communication medium.
- Procedural interoperability: a protocol and a syntactical form must be known and used for exchange.
- Operational interoperability: it refers to the activities related to the operation of a system in the context of other systems, e.g. doctrine governing the way the system is used. We differentiate the IT side of the operational interoperability (semantic interoperability between services) and the user side, i.e. how he understands information (sense-making and shared situational awareness) [EBR03], [SAS06], [WAR04].

An obvious solution for interoperability is to define an interface for each pair of communicating systems. However this can be achieved only for a small-scale system of systems, since for a large one, the number of interfaces necessarily leads to a very high cost, when feasible.

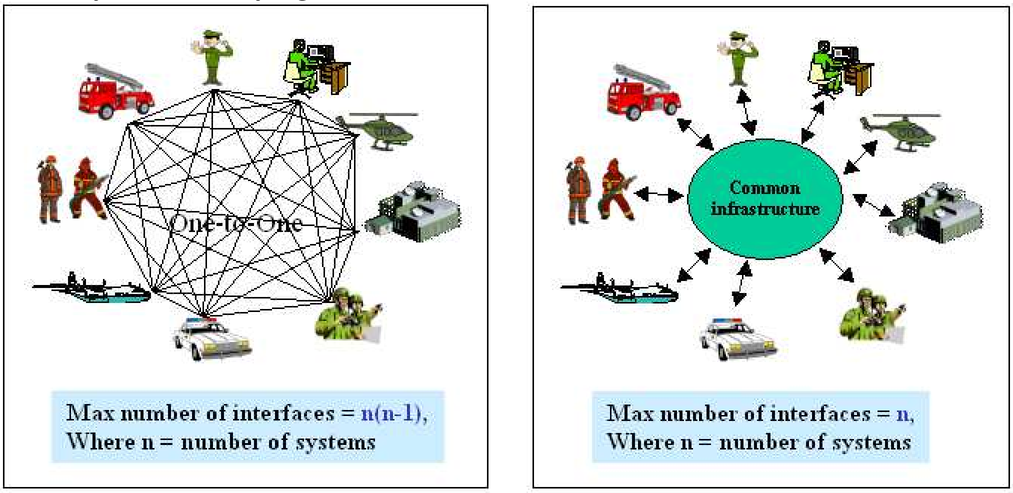


Figure 3: Common infrastructure decreases complexity.

Other solutions have to be put in place for interoperability, e.g.:

- Usage of a common technical infrastructure for physical interoperability: in that case the systems are no longer peer-to-peer linked, but each one is linked to the infrastructure.
- Usage of a common service-oriented infrastructure for procedural interoperability: this constitutes the current paradigm for information or IT-driven systems, and leads to service-oriented architectures. A “service repository” is expected to facilitate loosen the coupling between systems. It is empty when created and knows neither providers nor consumers. The service-providing systems access the repository to store the service definition in a neutral representation (location of the access point to invoke the service, service parameters, and quality of service). The service-consuming systems access the repository to get a service according to their need, and the invocation of the relevant service is then performed. This service repository plays the role of mediator and third party, and therefore enhances the security of the system by masking the service provider to the service consumer.
- Semantic issues for operational interoperability are more complex and not fully mature within the IT domain. They rely on the definition of common dictionaries (called “meta-data registry” in the US literature and “pivot model” in French) which are widely used for information systems and provide a public data model that allows communication between the different systems.

This multifaceted infrastructure which ensures general interoperability has to be included within the SoS as a new component with specific couplings with the relevant systems, and it has obviously its own life-cycle: in particular an adequate configuration management should be performed. Indeed a common infrastructure facilitates the architecture of the SoS at a given time, but not on the longer term: there is a necessary trade-off between immediate added facility (which increases short-term agility) and increased configuration management in time.

Coming back to our case study, the following table uses the coupling matrix and focuses on exchange and compatibility of services (which are mainly data in our case), indicating the versions of the exchanged services (service version on provider side, infrastructures version, service version on client side). This means that the system is currently running with the set of service versions. From this matrix it is straightforward to see the compliance requirements for a service: just look at the column to which the given service pertains. In the case of a system consuming services from several independent provider systems, the former must be compatible with all the interfaces of the latter through service version adaptation. For instance, the emergency call centre must accept the automatic call notification of any car. Hence the emergency call center needs to have the necessary adapters to translate the incoming data into its proper data model.

Automatic Call Notification	• Accident notification / versions (2.1, 2.0, 1.4)				
	Emergency call center	• Accident localization / versions (2.7, 2.0, 2.7) • Advised route communication / versions (1.9, 2.0, 2.6)	• Accident notification / versions (4.0, 2.0, 3.1)	• Plan transportation / versions (2.0, 2.0, 2.0)	
	• Accident scene reconnaissance / version (1.6, 2.0, 1.5)	**Medical emergency team / Ambulance**	• Medical data / versions (3.0, 2.0, 2.5)		
			Physician		
				Helicopter	• Patient vital data communication / versions (2.5, 2.0, 2.5)

Table 2: Data exchange and compatibility between systems.

Managerial independence, system's owner and manager issues

While the coupling of systems provides new capabilities and services, managerial independence of elements of the SoS means that each system is managed independently, including the evolution of the provided services, or the updating of data flows and interfaces. Each system evolves apart, depending on its owner's or manager's goals, needs and means. On the other hand, systems may operate for a long time. For instance, a car's owner can use its car for five or ten years, with little or no evolution of the embedded systems. Such asynchronous evolution issues are critical at the SoS level.

The coupling matrix provides a helpful insight to tackle such issues. Let us illustrate that on the EMS scenario. Table 3 lists the various systems' owners and managers.

Systems versus *owners and managers*	*Automatic call notification*	*Emergency call center*	*Medical emergency team / Ambulance*	*Physician*	*Helicopter*	*Hospital*
Owner	Customer	State or city	City or private company	Private	City or private company	City or private company
Manager	Car manufacturer	System provider	Ambulance manufacturer	System provider	Helicopter manufacturer	Hospital system provider

Table 3: Systems' owners and managers.

In addition to their variety, it should be noted that the systems themselves are part of different SoS. For instance, each car manufacturer manages its automatic call-notification system with its suppliers. This concerns the type of data, their semantic, syntax and format. Moreover, the evolution of the hospital system may be the result of economic constraints. If the hospital system provider updates the system, what are the impacts of this updating upon the interfaces with the other systems? Do these impacts necessarily imply evolution of the other related systems?

The coupling matrix helps answering such questions: if we read it as a process dependency matrix, the presence of many non-diagonal terms emphasizes tight coupling. On the contrary, a sparse matrix means a weak coupling. Thus the coupling matrix visually identifies sets of component systems whose collaboration is both essential and complex for the achievement of the emerging services. One of several possible SoS architecting processes can be defined:

1. Develop the scenarios describing the critical emerging functions.
2. Draw the resulting coupling matrix.
3. Identify the sets of strongly coupled systems (by permuting the systems, so as transform the coupling matrix into a block-diagonal matrix, as described in [MEI02]).
4. Once the critical sets are identified, adopt one or more of the following policies :
   - At least, each owner/manager of a component belonging to a critical set should be very cautious when designing a new version, and verify that interoperability is still ensured;
   - If it is possible, merge the management of the systems to ensure consistency of evolutions;
   - If it is possible, change the perimeter of the systems by merging them, from a technical point of view.

When looking at the EMS example for instance, we observe a weak coupling between the medical emergency team/ambulance and the other systems. Thus there are a priori few impacts following evolutions due to the car manufacturer. On the other hand, if the emergency call center system provider updates the system, there are many impacts on the medical emergency team/ambulance system, the helicopter system and the physician system. In this case, who is responsible for the evolutions of these systems? Who pays for them?

If the customer systems do not evolve at the same level, there is an asynchronous evolution of the SoS and an increased risk as to SoS capabilities. Lack of compliance means lack of interoperability, whence loss of emergent capabilities of the SoS. The asynchronous evolution issue is very important since owners and managers of the various component systems are different, with strong aims and constraints upon the systems. To deal with this problem, an independent organization, such as a state or an agency, may impose a globally planned and orchestrated evolution, resolving such asynchronous evolution issues.

Typically, when two systems are strongly coupled and exhibit weak coupling with the remaining systems, it could be appropriate to look at both systems globally and have a common organization level responsible for managing them. Such a question has its importance when addressing the communication infrastructure in a service-oriented architecture: who holds responsibility for this key asset of the SoS and manages its configuration, in adequacy with all other evolutions? A straightforward solution would be to have a contractor assuming integration responsibility for each subset of strongly coupled systems. Whether this can be applied to all problems is another question: on the one hand, it can be an advantage for Defense & Security SoS, or highly regulated SoS such as the air transportation and air traffic management SoS, but on the other hand it is an obvious barrier to spontaneous emergence of new behaviors.

Coupling matrix, asynchronous evolution and failure mode definition

The previous section hinted at how the coupling matrix could exhibit the impacts of asynchronous evolution, including emergent risks, as asynchronous evolution may degrade the whole SoS performance and safety. Indeed, as Levenson et al. [LEV06] write: “often, degradation of the control structure involves asynchronous evolution, where one part of a system changes without the related necessary changes in other parts. Changes to subsystems may be carefully designed, but consideration of their effects on other parts of the system, including the control aspects, may be neglected or inadequate”. Asynchronous evolution may trigger a failure of the interface between the related systems and imply by cascade effect a failure of the whole SoS. In this case, the new desired capabilities are not available, and worse, some critical component systems might have a failure that would not have occurred under stand-alone conditions.

A formal verification procedure based on formal techniques cited before can provide a static failure analysis of the SoS. The dependence coupling matrix is a useful representation for this, as it enhances readability of the compositional behavior. However it does not give any answer concerning dynamic failure analysis, since the dynamic environment of a behavior during execution cannot be captured unfortunately by static descriptions.

The only interesting answer, easily provided by our approach, is the search for a priori responsibilities between owners and managers in case of an identified failure. Indeed the incriminated service (either provided by a system, or arising as an emerging SoS service by combination of existing services) yields a set of potential responsible actors and resources, identified by reading the cells of the matrix. This raises the issue of responsibility transfers, which can be partially solved when an LSI (lead system integrator) or another appropriate risk-sharing or risk-assuming entity is designated. Back to our example assuming the SoS emerging services are in place, that allows a “real-time” reassignment of the evacuation means and the immediate health-caring environment following tele-diagnosis. If the patient’s condition unfortunately aggravates, who should carry the responsibility… and the consequences of potential sues?

An indication of the “manageability” of a safe SoS could be the number ofindividual interactions involving two different owners, and therefore requiring the establishment of a contract. This is enlightened by colored cells in the following table.
This rather obvious remark should be correlated with the impact analysis mentioned before, which relies on the connectivity degree of the various services: integrators carrying the responsibility should be designated a priori for the services with the highest connectivity index (the emergency call center in our EMS example).

Further extensions: towards a dynamical view of SoS management

Up to now we have dealt with static representations, isolating in time a specific view of the SoS. When addressing configuration management, a temporal analysis is more appropriate, especially for SoS which raise major agility issues: in that case each component (including the environment) is subject to radical changes in time that impact the whole SoS. This can occur at the (logical and physical) architecture level, at the mission level (evolving context of use and change of operating rules), at the organization level (vanishing and arising actors and/or contractors), etc. The instantaneous versions of the coupling matrix and its various enrichments can be assembled into a time-indexed bundle, and this new representation provides novel ways to tackle complex issues such as business strategic or acquisition issues: if system integrators are defined, their responsibility perimeter can be easily correlated to the evolution of the coupling within the SoS. For instance, the coloring mentioned in the previous section that helped seeing the connected business partners, defines in this higher-dimensional representation various colored subspaces (obtained by stacking the individual colored regions), which can be analyzed graphically very easily in terms of intertwining or connectivity.

Although this may seem a little far-fetched, it is a research direction worth to pursue, especially when one remembers that sustainability of complex systems is currently critical, as it concentrates the major part of the budgetary resources of the life-cycle for systems, and that we lack tools to manage such issues.

References

[Abrusci 95] Abrusci, V.M. Non-commutative proof-nets. In Girard, J.-Y., Lafont, Y.,and Regnier, L. (eds.) Advances in Linear Logic, pp. 271{296. Cambridge University Press. London Mathematical Society Lecture Note Series 222, 1995.
[DDSC07] Direction de la Défense et de la Sécurité Civiles, Protéger la forêt contre les incendies – Dossier de presse, 2007.
[EBR03] Evidence Based Research, Inc., Network Centric Operations Conceptual Framework, version 1.0, Evidence Based research, inc., 1595 Spring Hill Rd, Suite 250, Vienna, VA 22182, November 2003.
[LEV06] Leveson, N., Dulac N., Zipkin, D., Cutcher-Gershenfeld, J., Carroll, J. and Barrett, B, Engineering Resilience into Safety-Critical Systems, in Hollnagel, E., Woods, D. and Leveson, N. “Resilience Engineering, Concepts and Precepts”, Aldershot, Ashgate, 2006.
[MAI98a] Maier, M., Architecting Principles for Systems-of-Systems, Systems Engineering, vol. 1, n° 3, 1998.
[MAI98b] Maier, M., Architecting Principles for Systems-of-Systems. Electronic copy available at: www.infoed.com/Open/PAPERS/systems.htm.
[MEI98] Meinadier, J.-P., Ingénierie et intégration des systèmes, Paris, Hermes, 1998.
[MEI02] Meinadier, J.-P., Le métier d’intégration des systèmes, Paris, Hermes, 2002.
[MIL99] Robin Milner: Communicating and Mobile Systems: the Pi-Calculus, Cambridge Univ. Press, 1999, ISBN 0-521-65869-1.
[SAS06] SAS 050, Exploring New Command and Control Concepts and Capabilities, Final Report, prepared for NATO, January 2006.
[WAR04] Warne, L., Ali, I., Dopping, D., Hart, D. & Pascoe, C., The Network Centric Warrior: The Human Dimension of Network Centric Warfare, published by DSTO Information Sciences Laboratory, PO Box 1500, Edinburgh, South Australia, 5111, Australia, July 2004.
[Clark 99] Clark, T. & Jones, R. “Organisational Interoperability Maturity Model for C2.” Proceedings of the 1999 Command and Control Research and Technology Symposium. United States Naval War College, Newport, RI, June 29- July 1, 1999. Washington, DC: Command and Control Research Program (CCRP), 1999. <http://www.dodccrp.org/events/1999_CCRTS/pdf_files/track_5/049clark.pdf>(1999).