A Framework for Inter-Organizational Public Administration Network Services (July 2001)

Franco Arcieri, Roberto Giaccio, Enrico Nardelli and Maurizio Talamo

***Abstract*--The deployment of inter-organizational network services for the Public Administration is a challenging task, due to the broad range of strict requirements of both technical and organizational nature. In this paper we present a conceptual framework to describe application cooperation for inter-organization services that has already been adopted for the analysis and implementation of several existing Italian PA services.**

***Index Terms*--network services, application cooperation, traceability, security.**

I. Introduction

All e-government plans have the main goal to simplify interaction among citizens and the Public Administration by means of the development of inter-organizational network services where all involved Public Administrations cooperate to activate, synchronize and monitor all processes needed for a given service, and to guarantee the global coherency of data related to a given service in several Public Administrations.

There is now a big general effort to define common standards and protocols to represent and communicate structured information and services (XML, UDDI, SOAP) that simplify the definition of integrated network services. It is important however to observe that the problem to define inter-organizational services is primarily an organizational one, that is, even a complete solution to the problem of accessing services and exchanging information would not help the deployment of inter-organizational services unless we also consider problems like how such services are currently handled by the organizations, how the responsibilities for the services are assigned, how the organizations keeps track of the status of the services.

For instance, the analysis of an inter-organizational is simpler if there exist a single organization which has the capability to impose its technical solution to the other organizations, since in this case we basically have the same behaviour as a mono-organizational service; such assumption is reasonable where the organizations are hierarchically organized, and allows one to adopt a "technocratic" approach.

Much more difficult is the case where all involved organizations are peers: in this case we cannot impose a common set of protocols for the technical part, and processes and responsibilities are spread among the organizations with possibly different requirements on issues like authentication and security.

As a proof of this difficulty, United States of America, which are in general biased toward the adoption of innovative technologies, and which often have a high standardization of organizational processes, are quite cautious in publishing complex Public Administration network services, and tends to only offer simple informative services.

For instance, Rand Corporation [9] says:

"We are not aware of government agencies in the United States that are using the Internet to transmit sensitive personal data, e.g., tax, social security, and health information."

"This limitation on the current use of the net stems from our current inability to guarantee privacy, integrity, and authenticity. Obstacles to achieving the necessary level of security are NOT technical, however. Rather, they are institutional and organisational."

"These problems are compounded when data are combined from different agencies."

"Concerning the problem of new services arising by increasing the efficiency in the back-office, the following 4 types of architecture are possible for exchanging information among different organisations:

- free access in read/write to the database;
- trusted third party who takes care of logging the information fluxes;
- trusted third party who stores/forwards all information;
- publish and subscribe.

Case A corresponds to a merging among different organisations and can be used only when such a merging has been decided at all organisational levels. Case C implies the building of a new form of bureaucracy which may not be needed in developing the services. Case D should be limited to non-critical information (i.e. no privacy constraints, no economic content). Finally, Case B is the most broadly suitable and easily implemented possibility.

Technological tools like CORBA, DCOM and others are considered to be too complex in order to design (in a robust way) new services. On the other hand, XML standard is considered to be extremely important to follow and, with respect to government Agencies, it is considered to be a crucial point to develop a common dictionary to develop applications based on XML-structured messages."

Also, The Gartner Group in its analysis of e-government projects notices that:

"Developing a "government portal" to provide information on general issues is relatively simple and not expensive; a whole other complexity level is to provide applicative network

Manuscript received July 9, 2001.

F. A. Author is with the University of Rome "Tor Vergata", Via Orazio Raimondo,18 - 00173 Roma (e-mail: farcieri@gmx.it).

R. G. Author is with Unione Nazionale Comuni, Comunità Enti Montani (UNCEM), Via Palesto 30, 00185 Roma and with the University of Rome "Tor Vergata", Via Orazio Raimondo,18 - 00173 Roma (e-mail: giaccio@dis.uniroma1.it)

E. N. Author is with the University of L'Aquila, Piazza Vincenzo Rivera 1, 67100 L'Aquila (e-mail: nardelli@univaq.it).

M. T. Author is with the University of Rome "Tor Vergata", Via Orazio Raimondo,18 - 00173 Roma (e-mail: maurizio.talamo@aipa.it).

services, since it requires the modification of the Public Administration organizational models, creation of robust infrastructures for service access and monitoring…"

From the previous analysis, it is evident that a system of inter-organizational services for the Public Administration has to minimize impact over existing organizational and technical models, that is, the definition of such a system must be guided by relationships among organizations involved in the services. Along this line, we can cite the following points from the Italian e-government plan:

"The information systems of all Public Administrations must be connected by a peer-to-peer network, without hierarchies representing institutional or organizational superstructures."

"A national network interconnecting all Central and Local Public Administrations, based on the Internet model, and allowing to safely exchange peer-to peer applicative services among all Administrations.

In this paper we present a conceptual framework to describe application **c**ooperation for inter-organizational services that has already been adopted for the analysis and implementation of several existing Italian PA services.

In 1995 in Italy two of the authors started two important projects requiring application cooperation in the Public Administration, the Cadastral Municipalities Interchange System ("Sistema di Interscambio Catasto-Comuni", SICC), and the Mountain Information System ("Sistema Informativo della Montagna", SIM).

The SICC [1], [2], [13] is the Italian distributed cadastral system, and provide 8.000.000 cadastral and mortgage transactions per year to citizens and Local Public Administrations by means of distributed access points over the national territory.

The SIM [7] is a distributed network interconnecting more than 800 sites of heterogeneous Local (regions, forestry corps, mountain municipalities, mountain communities, national and regional parks) and Central (Agricultural and Forestry Ministry, Finance Ministry, Environment Ministry, National Statistical Institute, National Social Security Institute and others) Public Administrations. It provides a broad range of inter-administrational network services for territory management [5], automation of authoritative procedures, distributed sharing and update of geographical data [6]. Moreover, it gives a homogeneous access to Central Public Administration Services; for instance, it provides other distributed access points to the SICC. The SIM reaches about 10.000.000 citizens and covers 50% of the Italian territory with 4.000 municipalities.

In these two projects all involved organizations agreed on a common cooperation solution. Another system that is still currently being analysed and designed and presents the same kind of problems is the National Census Index ("Indice Nazionale delle Anagrafi", INA [8]), which provides distributed updates of all Italian citizens data from the municipalities to the central repository at the Home Office, notifying data changes to all registered Public Administration,

From the first analysis of these projects, it became clear that at that moment there were neither implemented nor widely accepted solutions to the design and implementation of such systems. It was also clear that a solution could only be found by first looking carefully at the following general issues:

- how the non automated Public Administration services worked;
- how the automated single Public Administration services worked;
- what we missed in developing the services in a multi-organizational context with respect to the simpler mono-organizational one;
- what technologies were available at the moment for application cooperation.

Note that these issues are independent on either more "technical" problems related to the actual services to be implemented, like the existing legislative framework, the precise specification of these services, or relevant "political" problems related to the formal agreements among involved Public Administrations on themes as sensitive data distribution, service access, responsibility.

Problems were further complicated by the fact that technologies, laws and political assets changed over time; for instance, on the technical side, the SIM topology changed from a network centred on the Central Public Administration, to a distributed one having Regions as main actors between Central and Local Public Administrations, and the SIM network had to integrate new Regional networks having different assumptions on important points like service levels and security; also, the network protocol changed from X25 to TCP/IP. Regarding the organizational side, several administrative tasks passed from the Central Administration to either the Regions or municipalities (for instance the cadastral tasks).

While designing and implementing these distributed systems for the Public Administrations we developed some original solutions [10], [11], [12] in some cases to overcome existing technologies limitations, in other to provide new technologies; of course, also these solutions changed over time as the technical environment and external requirements changed, but in general they proved to be general enough to withstand these big changes with only evolutionary modifications. Basically, we developed an hardware/software architecture that support application cooperation with the capability to trace inter-organizational processes, but also by improving security and user authentication; the key innovation in such architecture is that it is added on top of existing applications, requiring few modifications to existing applications and, more important, organizations.

Given the apparent generality of those solutions, and of course that fact that they worked well for these big projects,, we spent more efforts to define a conceptual framework to explain not only why they were adopted in the system, but also why we believe this set of functionalities is needed for application cooperation in important areas like as the Public Administration.

In the following, in Section I we discuss the main

problems arising when we have services involving several Public Administrations; then, in Section II we introduce the paradigms of ""intra-organizational validity" and "inter-organizational coherency", opposed to the stronger global validity paradigm found in centralized systems, and in general in mono-organizational services. These concepts allow us to introduce the concept of "cooperation backbone", that overcomes the limitations of standard distributed networks in the area of application cooperation. In Section III we show to what extent the cooperation backbone concept has been implemented in some existing Italian PA systems; finally, we present some conclusions and discuss some future improvements to the current implementation of the cooperation backbone.

II. Inter-organizational issues in PA services

The implementation of complex services for economical or administrative purposes in traditional centralized legacy environments, from mainframes to web systems, shows that, in order to guarantee that services are provided taking into account security, privacy, responsibility, and traceability, we need the following information:

- which are the "boundaries" of the informative system, how the service organizational units are interconnected;
- which applications are involved in the service processes;
- what connections to external systems exist, where are the possible intrusion points;
- which is the level of correctness and completeness of the requests issued by service users.
- for each service request, who made the request, from which workstation the request was issued, which service privileges are allowed to the user and the workstation, which security levels are needed for that request;
- for each failed service request, which data inconsistency, application malfunction or network component caused the problem.

The standard approach to the problem of obtaining these information uses heterogeneous tools and methodologies applied to different information system levels.

A. Network level

Protocols like SNMP monitor the network devices; specific router features like cryptography, proprietary protocols, secure lines protect privacy on network lines; proxies and firewalls limit unauthorized network accesses; processes analyse the log files of the network devices; on-line and off-line analysers determine and filter specific communication patterns.

B. At the operating system level:

Encrypted and shadowed password files, security levels C2 or higher, limit unauthorized operating system accesses; the removal of unneeded operating system services decreases the probability of attacks using service weaknesses; daemons responsible of periodic checking of possible operating system malfunctions limit host downtime and application malfunctions; accounting of operating system resource usage monitor misuse and control accesses to memory, CPU, disks, filesystems, network, processes, administration privileges, etc; programs that scan the operating system services look for possible attack points; journaled filesystems improve data integrity.

C. At the database level:

Referential integrity constraints in the database schema enhance coherency; views and tablespaces abstract and protect data; user privileges are assigned at the table level; unification of operating system users and database users limit unauthorized access; automatic data replication decreases database system downtime.

D. At the application level:

Strongly-typed languages, modules, namespaces, exceptions limit and control fallback of implementation errors; checkpoints in application log internal activities to the filesystem or remotely; unification of operating system and application users enhances security; security libraries, strong cryptography, public key methods, certificate authorities, smart cards improve privacy, user authentication and determination of responsibilities.

E. At the service access level

CORBA, RMI, HTTP protocols on top of encrypted protocols improve privacy; XML and SOAP standardize data structuring and access methods; logging of all relevant information for all transactions improves self-awareness.

Along this approach hardware and software companies, and consultant firms have proposed several solutions; so these solutions have always been characterized by low flexibility and have exploited only some of the techniques above; in general it is evident that this approach can present several problems, mainly due to its heterogeneity of the solutions it proposes.

The real limit of the approach become evident when we deal with inter-organizational services: whereas within a single organization a centralized control can integrate and organize the different levels abstraction of the solutions proposed toward a functioning system, the technical and organizational complexity of inter-organizational services adds to the overall complexity of the involved inter-organizational systems presenting different solutions regarding to networks, operating systems, applications and service access points. Moreover, we have the additional problem of not just make these systems communicate, but also integrate them on a higher level and make them cooperate maintaining global coherency.

In this context, a single problem among the ones cited above is seen only as a technological problem, and the solution is found by selecting some device able to extract and manage all information relevant to the solution of the problem itself.

What remains open, however, and this is fundamental in an inter-organizational service, is how to coherently integrate all information arising from the heterogeneous solutions to the various problems that can occur, determining the global status of the inter-organizational system with respect to the available services.

For instance (Fig. 1), consider a service consisting of three

distinct applications in charge of three different organizational units (U_a, U_b, U_c). In the case of simple organizations, the information sources needed to solve application cooperation problems are the network management, the system management and the three organizational units U_a, U_b, U_c responsible of applications a, b and c.

Service requests issued by the User have to transverse all the application path (U_a, U_b, U_c) to be accomplished. An interruption of the service due to technical or security problems in any point along the path should be monitored and handled

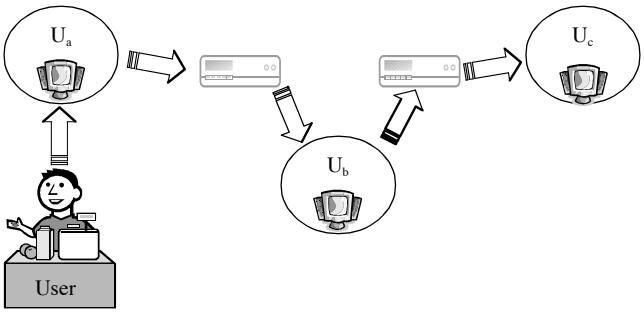

Fig. 1: a service consisting or three distinct applications.

If any problem occurs, in fact, we need first to correlate all log files and other information sources in charge to the organizational units, and then handle the overall problem; this can only be done by a higher organizational level with respect to the organizational units involved in the service: if these units are sub-units within a mono-organizational service this is possible, since this higher level can be realized internally by defining a unique point responsible of monitoring and handling all events arising from the correlation of the log files.

If a similar problem occurs in a true multi-organizational service, where units U_a, U_b, U_c belong to different and autonomous organizations, this centralization is not possible since the involved organizations should disclose the internal details in their log files, which is not possible due to leadership and privacy problems.

Note, however, that we cannot only consider the problem of merging different technologies and correlating log files information to solve possible problems, but we also need to handle the real aspects of cooperation among different administrations in order to handle cooperation events. This problem is structural and can only be solved by introducing a new organizational model that could take into account, for each involved organization:

- the specific constraints about privacy, security, technologies, etc.;
- how the applications in the organizations cooperate by means of cooperation events.

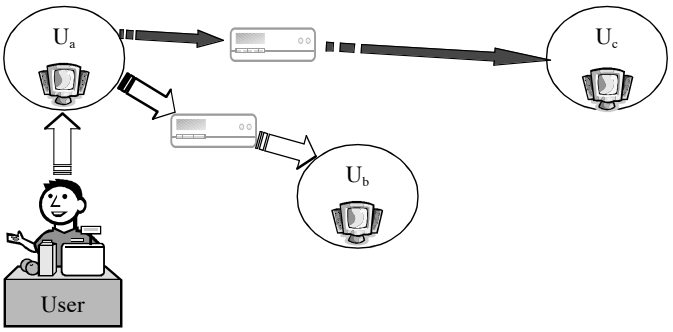

Fig. 2: a service request involving U_a and U_b and relevant for U_c

For instance, (Fig, 2), consider a service request issued from the User involving U_a ed U_b and having as a side-effect the update of an element of the information system of U_a. if the updated element is relevant outside U_a, like as Census information where U_a is a municipality and U_b is the Home Office and if U_c relies on Census information, U_c has to be notified of the update by means of a cooperation event.

The problem here is that it is not always possible to convince the organizations to "send" events to some event queue handler, since these handlers would become de facto supervisors of the internal services of the organizations; in this way the institutional and administrative autonomy of the organizations would be limited; this also implies that the organizations would have to agree with the supervisor on any design, change or evolution in their internal systems.

Hence, we have to determine an intermediate way to correlate the information involved in a multi-organizational service that:

- guarantees the correct handling of problems;
- monitors services for cooperative events;
- uses a minimal set of information from the involved organizations, possibly only information related to the cooperation among organizations.

This model is the *cooperation backbone*; in the following section we present the theoretical ground of this model.

III. The Cooperation Automata

In order to present the main guidelines of our solution, we focus the attention to the very basic problem of reducing and tracing errors in automated services; although in many cases we are interested to monitor other aspects of the service like users, data accesses and many other details, this is indeed a very meaningful setting due to the obvious observation that if there were no errors at all we would rarely pay attention on such details; unfortunately in real systems a lot of errors can occur, like communication errors, data inconsistency, misinterpretation of data, human errors; so the need to know all details of the service arise from the possibility of errors.

A first important observation about this is that often we do not distinguish two different requirements: reducing the errors in the system and keeping track of these errors. Of course, reducing errors is an important goal, not only in itself, but also since it reduces the need for the latter requirement; however, we must convince that errors are not always completely eliminable, at least not in systems ultimately

relaying on humans beings to accomplish complex tasks.

Hence, the second goal becomes important, and systems must implement facilities to trace the internal working in order to:

- determine if an error has occurred;
- determine where the error has occurred;
- determine who is responsible for the error.

In general while designing a system, the efforts tends to concentrate on the first requirement, for instance by defining a strong infrastructure for data communication among automated processes, or by relying on robust technologies for user authentication, privacy, etc.

It is quite natural, however, that possibly completely different strategies and technologies can be needed in order to keep track of errors. This misconception is shown, for instance, by the fact that in some documents where solutions as "publish and subscribe" and "log file analysis" are presented as possible alternative design choices, whereas the former defines a data communication infrastructure while the latter defines a solution to the problem of identify and localize errors.

Along this way it is straightforward that the problem of tracing errors, and the related and more general problem of monitoring system services, could either be handled by an independent trusted layer or by designing a system that takes into account the problem by itself. Apparently this latter solution, designing a communication system that both handles transactions and keeps track of all transaction details, is a better one. It must be noted, however, that this is not always possible or practical; for instance the data communication can be bound to a legacy system that is difficult to substitute or modify. More radical problems arise in systems where several organizations have to cooperate to accomplish the service tasks:

- organizations could have different data communication and service implementation solutions, and cannot agree on a common standard;
- organizations have different strategies on security and privacy;
- organizations handles similar data in different ways, and there is the need to correlate these data;
- even if the service if composed by services of the involved organizations, in no organization there is a complete knowledge about the compound service.

Hence, the first solution could be feasible for simple mono-organizational services, but the first one is better suited to complex inter-organizational services.

In the following we substantiate this claim by using some simple arguments from the Finite State Automata (FSA) theory, first by modelling the problem of tracing errors within a single organization, then by extending it to inter-organizational services.

1) Single organization

Suppose we have a mono-organizational service; in its simpler form, the service consists of several steps, each step linked to some possible successive steps; this setting fits easily in a document workflow automation system, where we define a set of states we define possible outcomes from a state.

This can be modelled with a finite state automata A with states set $Q_A = \{q_1, q_2, ..., q_n\}$ and transitions set T_A $\{(q_i, c_j) \rightarrow q_k)\}$; a document passes from state q_i to state q_k if the character c_j is read.

Hence, a document workflow corresponds to a *computation* C of the automata, that is, a sequence of alternating states and characters starting end ending with a state $(q_{i_1}, c_{j_1}, ..., q_{i_{m-1}}, c_{j_{m-1}}, q_{i_m})$. A *computation step* is a triad $(q_{i_h}, c_{j_h}, q_{i_{h+1}})$, and is said to be *valid* if $(q_{i_h}, c_{j_h}) \rightarrow q_{i_{h+1}}$ belongs to T_A. A computation is *valid* for A if all computation steps are valid; see Fig. 3 for an example.

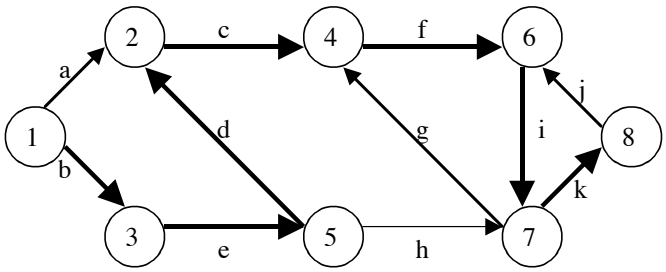

Fig. 3: the valid computation (1, b, 3, e, 5, d, 2, c, 4, f, 6, i, 7, k, 8).

We model errors in the system by defining a second real-word FSA B with $Q_B = Q_A$ that acts as a generator of computations for the first one; this automata is probabilistic, and has a probability $P(q_i, c_j, q_k)$ for each state transition from q_i to q_k and each character c_j, with the obvious constraint $P(q_i, *, *)$ summed for all outgoing states and characters is 1. The automata B generates computations C that model possibly bad document workflows; the automata A accepts C only if it is a valid computation; for instance, the FSA in Fig. 1 rejects computation(1, a, 3, e, 5).

One could object that it is easy to design a system that merges A and B into a single application layer both responsible for execution and validation of services. As we will see, the same cannot be done for multi-organizational services.

2) Multiple organizations

Starting from the previous automata A and B, a service involving n organizations can be modelled by partitioning the state set Q_A into several subsets Q_{A_1}, Q_{A_2}, ... , Q_{A_n} , each belonging to a different organization.

We partition the set of possible transitions into m+1 subsets T_{A_0}, T_{A_1}, T_{A_2}, ..., T_{A_n}, where T_{A_i} contains all transitions $(q_i, c_j) \rightarrow q_k$ s.t. q_i in Q_{A_i} and q_k in Q_{A_i} i = 1, ..., n, and Q_{A_0} contains all other transitions; we can now define n automata $A_i = (Q_{A_i}, T_{A_i})$, i = 1, ..., n and an automata $A_0 = (Q_A, T_{A_0})$; see Fig. 4 for an example. We call the automata A_0 responsible of checking inter-organizational transitions the *cooperation automata*.

Furthermore, we partition each computation C into n+1 subsets C_0, C_1, C_2, ..., C_n, with C_i containing the set of maximal sub-computations of C with all states belonging to Q_{A_i}, i = 1, .. , n, and C_0 containing all other maximal sub-computations. We say that a computation C is *locally valid* for A_i if all computations in C_i are valid for the automata A_i, i = 1, ..., n,, and we way that C is *coherent* if C_0 is valid for

A_0. For instance, for the FSA in Fig. 1 the non-valid computation (1, a, 3, e, 2, c, 4), split into {(1, a, 3)} inside A_1, {(2, c, 4)} inside A_2 and {(3, e, 2)} inside A_0, is locally valid for both A_1 and A_2, but not A_0, so it is locally valid but not coherent.

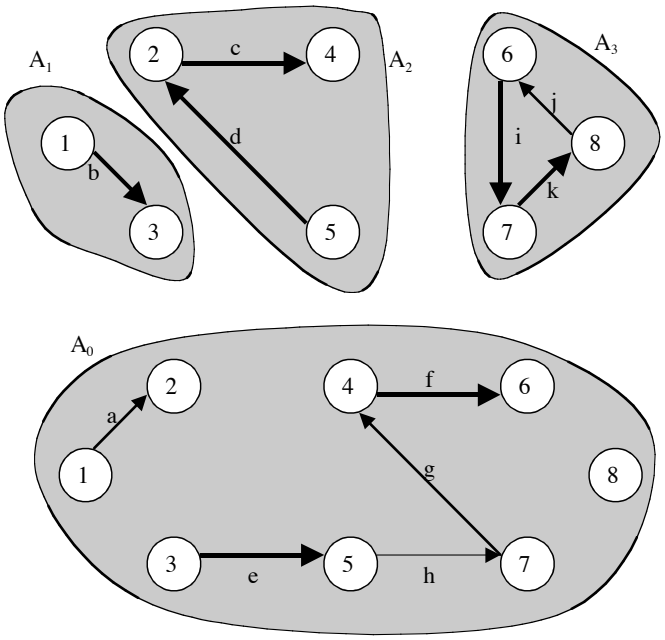

Fig. 4: an FSA split among three organizations into automata A_1, A_2, A_3 and A_0 (the set of states is shown twice for sake of clarity); the computation (1, b, 3, e, 5, d, 2, c, 4, f, 6, i, 7, k, 8) is split into {(1, b, 3)} inside A_1, {(5, d, 2, c, 4)} inside A_2, {(6, i, 7, k, 8)} inside A_3 and {(3, e, 5). (4, f, 6)} inside A_0, all locally valid.

Note that a valid computation corresponds to a workflow that is correct for all organizations involved; a workflow can be not correct, even if it is correct for all organizations, if there is any error while passing from one organization to another; from the partitioning of the initial FSA the following fact trivially holds:

A computation is valid for Q_A if and only if it is locally valid for all A_i, i = 1, ..., n, and is coherent.

This simple structural property allows us to clearly divide the validation process into two different tasks: testing validity internally inside each organization, and testing coherence; this latter process is exactly the additional checking needed for application cooperation; in general checking internal correctness is an easy task, since this can be done in a centralized way, with all details decided internally by the organization itself. On the other side, checking the coherency can be, and is, a very difficult task [4].

Note that here we assume that the FSA A is known in all its details; this assumption allowed us to present this FSA model in a plain way, but is not reasonable in practice: when designing an inter-organizational process we often just cannot know all internal details of the transactions occurring in a given organization; all we can get is a certificate that the process has been performed correctly, so the multi-organizational problem is better modelled by an unknown FSA A split among several organizations; we cannot monitor the workflow internal to each organization, but the organizations can provide us a certificate of validity: this certificate allows us to assign responsibility to an organization into which an error occurred, and delegate to the organization itself the task to precisely identify the internal faults.

Hence, in order to guarantee that all things have been performed correctly, we have to accomplish the following tasks:

- we have to receive a sort of "certificate of validity" from each involved organization for all service processes; this can be done by assuming that any transaction which is not valid is be flagged by the organization;
- we have to monitor transitions in T_{A_0}; doing this implies having a trusted infrastructure that checks that each transition from an organization to another one is correct. This can be done, for instance, by scanning the log files of two connected organizations and checking that they are consistent, or by monitoring the traffic among the two organizations and performing consistency checks in real-time.

This behaviour defines a conceptual framework into which discuss and define different solutions for providing multi-organizational services; it is worth noting that we cannot check less than what we do and guarantee consistency; on the other side, checking more does not give advantages.

In Section II we showed that a service involving two different organizations could possibly need to keep track of transactions among these organizations to perform notification of events to other organizations. It is easy to see that the computations validated by the cooperation automata describe all the events possibly interesting other organizations; thus, an implementation of the cooperation automata would have knowledge of all the information needed to perform notification of events.

IV. The Cooperation Backbone

From the discussion in Section I, the problem of application cooperation could not be solved by the straightforward introduction of technologies for handling events or errors because of the heterogeneous nature of these possible events or errors; we also showed that there are cases when we have to keep track of what is going on between the organizations to be able to notify events related to the service. Also, application cooperation could not ask to the involved organization to disclose more data than the minimal set needed for application cooperation. In Section II, we introduced the cooperation automata and defined this minimal set with respect to the problem of errors identification, and showed that this minimal set is also sufficient to the purpose of event notification.

In this section we present the notion of *cooperation backbone* and *active tracing* as practical solutions that summarizes all the presented arguments.

A system of inter-administrative services is built on a set of primary information sources (Census, Cadastral, etc) that are always in charge of a Public Administration. This

responsibility is defined in terms of monitoring coherency, quality and accessibility of these informative sources. On these sources we have a set of administrations without hierarchical relations that use the information to develop services for their specific purposes; this holds, to the best of our knowledge, for all administrative processes in Italy; this concept is often improved by initiatives of organizational or administrative nature which define a cooperation and standardization model for such services.

A *cooperation service system* is a set of services from a set of cooperating organizations that use the same primary informative source. Each organization handles its subset of services autonomously. Given a cooperation service system, its multi-organizational aspects are governed by its *cooperation backbone*, which:

- monitors and understands all inter-organizational transactions;
- detects possible service coherency errors;
- notifies involved administrations of coherency errors;
- notifies organizations of events;
- monitors notifications.

For sake of convenience it also adds the following features:

- correlates transactions among organizations into a unified view;
- monitors resources involved in inter-organizational transactions;
- guarantees security to inter-organizational transactions;
- can identify the specific workstation issuing a service request.

In order to monitor, understand and correlate multi-organizational transactions, the cooperative backbone relies on *active tracing*; this technique allows the cooperative backbone to be independent on the technologies used for inter-organizational transactions for network, operating systems, database, application and service access. Active tracing consists of scanning in real time all outgoing and incoming inter-organizational transactions and extracting all and only the information needed for maintaining service coherence. This scanning requires that the needed information can be accessed, so for instance, it cannot be issued after on encrypted transactions; however, it has been proved general enough to trace HTTP/HTML and HTTP/XML transactions, many legacy systems and even VT400 transactions from a terminal emulator; within these requirements, active tracing does not require any modification to existing systems.

The active tracing, as all the cooperation backbone is bound to the administration responsible of the primary information source, which must guarantee the correct use of the extracted information.

As an example we now present the application backbone of the SIM System, the SICC being similar from the application cooperation point of view; a conceptual view is shown in Fig. 5.

In the SIM all inter-organizational transactions are monitored by *certification probes*; these are hardware or software devices that intercepts outgoing and incoming communications between the Public Administrations involved.

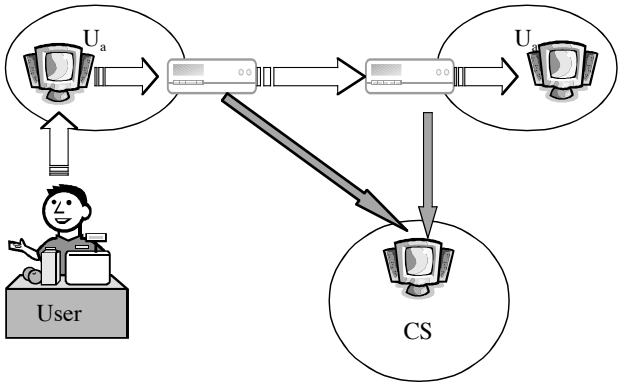

Fig. 5: the SIM cooperation backbone

These probes assume that a SIM transaction always have a standard header describing all service parameters; only this header was used in the probing process.

Certification probes can be configured remotely as new services are defined; at the beginning were mainly intelligent packet sniffers; then they evolved into passing probes, acting as application gateways.

The responsibility of the system, and hence of the correct use of traced data, is in charge of the Agricultural and Forestry Ministry.

All traced data are sent to a central system at the SIM Service Centre (CS) of the Agricultural and Forestry Ministry, and stored on a dedicated database. An application server, the *certification server*, continuously check incoming data for service coherency.

From a high level point of view, in the SIM architecture the certification probes feed the Service Centre database with all computations corresponding to the inter-organizational part of the service workflow. The application server implements the cooperation automata.

Of course the SIM architecture is much more complex than this, and the SIM cooperative backbone has much more structure than that in Fig. 5. For instance, the network part as several intermediate levels that abstract service access specific protocols and network addresses. Furthermore, there is a whole part of the certification server responsible for correlating data from different organizations that maintains locally a minimal set of keys from all the organizations [3], [4].

V. Conclusions and Future Work

We presented a model for multi-organizational application cooperation, the cooperation backbone, that is both sound from the theoretical and modeling point of view, and technically feasible, having been used as the ground base for implementing, in several projects, the state of the art of application cooperation for the Public Administration in Italy. The model clearly separates and structure the aspects of application cooperation relative to intra-organizational privacy and security, responsibility of sub-services within an organization and overall service responsibility, and provide a feasible layer of abstraction above network, operating system,

database application and service access layers whose unique purpose it to provide an adequate set features to implement application cooperation.

Moreover, the model permits a gradual integration of systems and is compatible does not require modification to legacy systems, feature that has been extensively used in the implementation of the Italian systems SIM and SICC.

Of course, a lot of work still needs to be done; in particular, even if the proposed model only prefers inter-organizational transactions to be traceable by means of an header containing all relevant information, some standardization in the formats used for application cooperation would be useful; in particular, the use of XML and SOAP for service access seems preferable, also because the possible lack of privacy caused by the open XML structure could be overcome by the channel security features of the current implementations of the cooperation backbone.

As a final remark, beyond the improvements that can be done to the current implementations of the cooperation backbone, the current open problem is how to merge several cooperation backbones, that is, how to integrate heterogeneous cooperation solutions among organizations involved with services from different primary data sources: this is a problem of higher complexity that still has to be solved at the modelling and technological level. We are currently designing an evolution of the cooperation backbone found in the SIM for the INA System, where we hope to extend both the model and the its implementation to provide an extendible and standardized layer for application cooperation.

REFERENCES

[1] F. Arcieri, C. Cammino, E. Nardelli, M. Talamo, A. Venza: The Italian Cadastral Information System: a Real-Life *Spatio-Temporal DBMS, Workshop on Spatio-Temporal Database Management* (STDBM'99), Edinburgh, Scotland, U.K., Sep.99, Lecture Notes in Computer Science vol.1678, 79--99, Springer-Verlag.

[2] F. Arcieri, C. Cammino, E. Nardelli, M. Talamo, A. Venza: Italian Cadastral Data Exchange System, *GIM International*, Dec.99, 13(12): 6--9.

[3] F. Arcieri, E. Cappadozzi, P. Naggar, E. Nardelli, M. Talamo: Access Key Warehouse: a new approach to the development of cooperative information systems, *4th Int. Conf. on Cooperative Information Systems* (CoopIS'99), Edinburgh, Scotland, U.K., 46--56, Sep.99.

[4] F. Arcieri, E. Cappadozzi, P. Naggar, E. Nardelli, M. Talamo: Coherence Maintenance in Cooperative Information Systems: the Access Key Warehouse Approach, *International Journal of Cooperative Information Systems*, to be published on Sep.01.

[5] F. Arcieri, E. Cappadozzi, E. Nardelli, M. Talamo: Geographical information systems interoperability through distributed data exchange, *1st International Workshop on Databases, Documents, and Information Fusion* (DBFusion'01), Magdeburg, Germany, May 01, Preprint n.8/2001, Fakultaet fuer Informatik, Universitaet Magdeburg.

[6] F. Arcieri, E. Cappadozzi, E. Nardelli, M. Talamo: Distributed territorial data management and exchange for public organizations, *3rd International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems* (WECWIS'01), San Jose, Ca., USA, Jun.01, IEEE Computer Society Press, 2001.

[7] F. Arcieri, E. Cappadozzi, E. Nardelli, M. Talamo: SIM: a working example of an e-government service infrastructure for mountain communities, *Workshop On the way to Electronic Government* (DEXA-eGov'01), associated to the 2001 Conference on Databases and Expert System Applications (DEXA'01), Sep.2001, Munich, Germany, IEEE Computer Society Press.

[8] F. Arcieri, M. Talamo, "Il modello di funzionamento del dominio di servizi dell'INA," unpublished.

[9] Rand Corporation, AIPA Meeting: the internet and Public Administration in Italy, summary report available at http://www.aipa.it/english[4/internet[4/index.asp

[10] M. Talamo et al., *Apparatus and method for monitoring and interpretation of application protocols for network data transmission problems*, Patent EP0960506.

[11] M. Talamo et al., *Apparatus for control and certification of the delivery of goods object of electronic commerce and for the concurrent control and certification of the execution of the related payment*, Patent EP1104574.

[12] M. Talamo et al, *Network access control device through fast recognition of application frames*, patent WO0010304.

[13] M. Talamo, F. Arcieri, G. Conia, E. Nardelli: SICC: An Exchange System for Cadastral Information, *6th Int. Symp. on Large Spatial Databases* (SSD'99), Hong Kong, China, Jul.99, Lecture Notes in Computer Science vol.1651, 360--364, Springer-Verlag.