

DocFlow: analysis, monitoring, and optimization of Web documents and services

Anca Muscholl Serge Abiteboul Albert Benveniste

9th May 2006

Abstract

DocFlow builds on the *Active XML* formalism for data exchange across peers on the Web, developed by one of the participating teams. The project aims at achieving convergence of data and workflow management over the Web through the concept of distributed active documents. We believe that this opens new perspectives for distributed e-management in the context of a global economy.

Data exchange, web data and services, workflow, distributed systems

Contents

1	Project objectives	3
1.1	Context and motivations	3
1.2	Novelty, high level objectives, and key expected results:	3
2	State of the art	4
2.1	Related work	4
2.2	Background from the teams	5
3	Detailed project description	6
3.1	Overall research objectives	6
3.1.1	A brief introduction to AXML	7
3.1.2	Research problems	8
3.2	Detailed work programme	12
4	Deliverables and responsibilities	15
5	Detailed expected results and criteria for success	16
6	The teams and their qualification	17
6.1	The Liafa team and its key persons	17
6.2	The Gemo team and its key persons	17
6.3	The DistribCom team and its key persons	18
6.4	Complementarity of the teams and cross-disciplinarity of the consortium	20
6.5	International collaboration	20

1 Project objectives

1.1 Context and motivations

Since the 60's, the database community has developed the necessary science and technology to manage data in central repositories. From the early days, many efforts have been devoted to extending these techniques to the management of distributed data as well, and in particular to its integration, e.g., [28, 54, 50]. However, the Web revolution is setting up new standards, primarily because of (i) the high heterogeneity and autonomy of data sources, (ii) the increasing complexity and richness of data, and (iii) the scale of the Web and the diversity of interaction among its users.

On the other hand, the increasingly global economy calls for tighter integration of global enterprises and OEM-supplier chains. At the same time, global enterprises and OEM-supplier chains are becoming more and more widely distributed and OEMs get constantly seeking for best suppliers. Such distributed workflow activities must rely on a light weight infrastructure, yet capable of providing predictable, safe, and secure workflow execution. Recently, standard languages for service workflow have even been proposed such as IBM's Web Services Flow Language [56] or Microsoft's XLang [57], which converged to the BPEL4WS proposal [15] and subsequently WSCDL proposal [43] for choreographies. A recent overview of existing work can be found in [33]. The implementation of orchestration and choreography description languages raises a number of difficulties related to efficiency and clean semantics and reproducibility of executions that are impairing their industrial acceptance.

A serious shortcoming of approaches to Web Service orchestration and choreography is that they mostly abstract data away. Symmetrically, current approaches to Web data management typically based on XML and XQuery rely on too simplistic forms of control.

*We believe that time has come for a convergence of sophistication in terms of control and richness in data, for workflow and data management over the Web. We believe that **active Peer-to-Peer XML-based documents** provide the basis for an adequate infrastructure for this. The overall objective of this project is thus to propose such an infrastructure and study its mathematical foundations.*

1.2 Novelty, high level objectives, and key expected results:

- (a) Ensuring convergence of data and workflow management with a focus on Web information management.
- (b) Defining an infrastructure of active Peer-to-Peer documents able to perform stateful distributed activities.
- (c) Providing Web compliant alternatives to existing distributed database technology, making use of no locking mechanism.

- (d) Developing a technology for Web services orchestrations and choreographies, based on the central notion of document.
- (e) Developing models and approaches to handle performance, monitoring, and other Quality of Service aspects, for our infrastructure of active Peer-to-Peer documents.
- (f) Developing novel techniques to strengthen some recognizedly weak aspects of Web Services technology regarding security.
- (g) Establishing all the above on a formally sound basis.

In the rest of this proposal we provide technical elements in support of our approach, and translate the above high level objectives into scientific problems for research.

2 State of the art

2.1 Related work

The DocFlow project relates to several different research areas and uses background from various communities. We briefly review these.

Distributed systems, P2P and Distributed query optimization. In the context of distributed data management, distributed query processing has been studied since the early days of databases [40, 50], and in particular in the context of mediator systems [30, 52] and P2P environments [5, 24, 31].

Peer-to-peer. This term refers to a class of systems and applications performing a function using distributed resources, with no centralized control and a dynamically evolving set of peers. Together, peers may produce computing power as in, e.g., seti@home, or storage space as in, e.g., Napster [44] or KaZaA [36]. Distributed hash tables, e.g., [47] is an example of popular P2P technique. Peer computing is gaining momentum as a large-scale resource sharing paradigm by promoting direct exchange between equal peers, see [36] or [41]. In this project, we propose a system where interactions between peers are at the core of the data model, through the use of service calls.

XML documents with embedded Web services calls. Service calls in semi-structured data have been considered in the context of Lore and Lorel [32]. Other systems recently proposed languages based on XML or other documents with embedded calls to Web services [16, 34, 46]. AXML is more powerful as it provides means of controlling and enriching the use of Web service calls for data and workflow management purposes, in a distributed setting. Also, AXML is a continuation of the work on *ActiveViews* [2]. The main differences with ActiveViews is that AXML promotes peer-to-peer relationships vs. interactions via a central repository. The activation

of service calls is also closely related to the use of *triggers* [51] in *relational databases*, or rules in *active databases* [53]. Active rules were recently adapted to the XML/XQuery context [13]. A recent work considered firing Web service calls [14]. AXML goes beyond those by promoting the exchange of AXML data.

Data integration systems. These typically consist of data sources, which provide information, and of mediators or warehouses, which integrate it with respect to an integration schema [25, 38, 40]. AXML takes a hybrid path between mediator systems (the integration is virtual) and warehouses [29] (all data is materialized). Mappings between data sources are captured in AXML by service calls embedded in the data.

Service composition and workflow. The integration and composition of Web services has recently been an active field of research [18]. Standard languages for service workflow have been proposed such as BPEL [15], and the WSCDL proposal [43] for choreographies. A recent overview of existing work about service composition can be found in [33]; therein, services are communicating Mealy machines together with input/output signatures on messages (given by XML Schema types).

Mobile code. Mobile codes are programs that use mobility as a mechanism to adapt to resource changes, cf. the Join-Calculus [35] and the Sumatra language [7]. In our case, peer to peer architectures and asynchronous communication are used; also active documents are exchanged, but our active documents are more restricted than general code.

Distributed monitoring of networked systems. Attention has been paid to dealing with large distributed systems that cannot be monitored as a whole, for reasons of size [9, 49, 12, 61, 26, 17]. The work [23, 10, 22] deviates from the above by explicitly handling available concurrency in large distributed systems; unfoldings and similar techniques are used in combination with modular algorithms, resulting in a supervision architecture that is itself distributed.

2.2 Background from the teams

The project builds on the following background from the three teams:

- Active XML (AXML in short) [8] has been proposed and developed for Web scaled data management by Serge Abiteboul and the Gemo team¹. AXML is a language that leverages Web services for distributed data management and is put to work in a peer-to-peer architecture. AXML documents are XML documents with embedded calls to Web services. See Section 3.1 for an introduction to AXML.

¹gemo.futurs.inria.fr

- Algorithms are needed to monitor and control Web scale networked systems, to evaluate their Quality of Service (QoS), and to negotiate contracts. These techniques can be seen as contributing to the realm of “autonomic computing”, i.e., the technology enabling autonomous and adaptive operation of Web scale distributed systems. *True concurrency models for distributed systems and associated algorithms* have been developed by the DistribCom team² headed by Albert Benveniste, with applications to fault management and alarm correlation in telecommunications networks [23, 10, 22]. These algorithms use net unfoldings, event structures, or some special classes of graph grammars to address dynamism.
- The games paradigm appears naturally in the analysis of open systems, i.e., of systems that interact with an environment or with external users. While 2-person games (with regular winning conditions) are pretty well understood, the distributed case is much more challenging, [45, 37, 39]. *Games in the context of AXML* materialization have been investigated by Anca Muscholl (Liafa)³ in cooperation with the Gemo group [42, 6]. Victor Vianu and the database group at UC San Diego have worked recently on the *verification of compositions of Web service peers* that interact asynchronously by exchanging messages [20, 21, 19].

Cooperation between these three teams was initiated by the INRIA Cooperative Research Action called ASAX.⁴ In ASAX, first studies on unfolding semantics of AXML were initiated.

3 Detailed project description

3.1 Overall research objectives

Our approach builds on the novel concept of *Active XML (AXML)* documents. AXML was originally intended for data management. As such it relies on simplest mechanisms for its control by putting emphasis on document manipulation. When restricted to stateless documents, AXML peer systems can be given a clean and unambiguous semantics, not subject to Web nondeterminism other than failure to answer.

Upgrading AXML to the area of distributed workflow management requires being able to handle documents with states and richer control. This requires revisiting how AXML queries should be answered, which in turn calls for advances in the semantics of AXML peer systems. In addition, optimization and monitoring of AXML peer systems requires non-trivial distributed algorithms.

In this section, we present the technical elements supporting our research and we translate the high level results presented in Section 1.2 into scientific problems for study in this project.

²<http://www.irisa.fr/DistribCom/>

³<http://www.liafa.jussieu.fr/>

⁴<http://gemowiki.futurs.inria.fr/twiki/bin/view/Gemo/AsaxWeb>

3.1.1 A brief introduction to AXML

XML [58] is a data model and representation format for semi-structured data [4], that has been proposed for data exchange between remote applications, notably over the Web. XML consist of ordered trees where node labels provide the semantics of the data element. XML comes equipped with high level query languages, e.g., XQuery [60], and flexible schema languages, e.g., XML Schema [59]. XML data publication and access are facilitated by *Web services*, which are network-accessible programs taking XML parameters and returning XML results. The WSDL [55] and SOAP [48] standards enable describing and calling these remote programs seamlessly over the Internet.

Peer-to-peer architectures propose an alternative to centralized ones, much more in the distributed and autonomous spirit of the Web. They are already spreading, notably in the context of file-sharing [36, 27, 11], primarily because they scale much better than centralized systems to very large number of machines. Peer-to-peer architectures capture the autonomous nature of the systems participating in them, and the ability of these systems to act both as producers of information (i.e., as servers) and as consumers of information produced by others (i.e., as clients).

Active XML (AXML in short) [8] is a language that leverages Web services for distributed data management and is put to work in a peer-to-peer architecture. *AXML documents* are XML documents with embedded calls to Web services. Such documents are enriched by the results of invocations of the service calls they contain. Since service invocations rely exclusively on Web services standards, AXML documents are portable and can be exchanged. The AXML model also defines *AXML services*, which are Web services that exchange AXML documents. The ability of participating systems to exchange AXML documents leads to powerful data-oriented schemes for distributed computation, where several systems dynamically collaborate to perform a specific data management task, possibly discovering new relevant data sources at run-time.

AXML documents. Figure 1 shows a portion of a simple AXML document used in a manufacturing information system.

When activated, a service call embedded in an AXML document invokes the corresponding service (a query), and the result of this invocation (a forest) is inserted in the document. This process is termed *materialization*. Technically, the service invocation is done using the SOAP protocol, based on the attributes of the *call* element. The sent SOAP message also contains the values of the parameters (i.e., the children of the *call* element). The service answers this request with a SOAP message, which contains the result of the invocation, or an error message. If no error occurred, the result is inserted in the document, *in place of* the service call element. Figure 2 shows our sample document after the materialization of the call to **Checkstock**. This time, we use a tree representation instead of the less intuitive XML syntax.

```

<warehouseStock>
  <table quantity="4">
    <minimumstock q="5">
      <local quantity="3"/>
      <status mode="critical"/>
      <sc Stock@Ext></sc>
      <table parameter="arg"/>
      <result quantity="1"/>
      ...
    </table>
  </warehouseStock>

```

Figure 1: *An Active XML document.* A company has a factory and a warehouse. The document giving the information at the warehouse describes in particular the quantity of tables in stock at the warehouse (here 3), and the quantity in stock at the factory (here 1). This last number was obtained by the remote invocation of the service call `Stock@Ext`. The codes of service calls is not shown here. In particular, there is a continuous service call that monitors the warehouse, detects when there is shortage for some piece and then reacts by sending an order to the factory (to produce the piece).

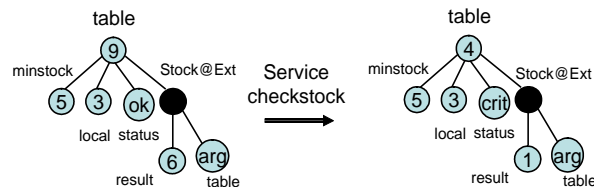


Figure 2: *Our sample document after the materialization of the call to Checkstock.* The initial result of `Stock@Ext` is 6, which means that there are more than 5 tables (6 in factory+3 local) hence the status is ok. The next check reports that the number of tables in the factory is 1, hence there are less than 5 tables and the critical status is triggered. Next, an order for table will be sent to the factory.

3.1.2 Research problems

We discuss research problems and indicate, for each of them, the existing background and challenges.

Convergence of workflow and XML document management. Our vision is that the convergence of workflow and XML document management is now required, and that it will occur by building around the concept of *document*. We believe that documents will receive easy acceptance as a paradigm for global enterprises and more generally B2B interactions. We believe that, for those application areas, data (in the form of documents) cannot be separated from control but should rather deeply interact with each other. Our central idea is that a clean concept of active document operating on a peer-to-peer architecture

is the adequate solution.

We see AXML as a first and fundamental step toward achieving this. Clearly, AXML was designed for data, not control management. Therefore, the first set of questions we need to investigate is: How far can we address workflow through AXML systems? Is document based workflow achievable?

Next questions are: Do we need extensions of the AXML language, or do we only need refined semantic studies and modifications of the AXML engine? Clearly, we favor not extending the formalism itself; if such an extension is still needed, do we need just syntactic sugar in the form of macros? Or alternatively, do we need deeper extensions?

Problem 1 (convergence workflow/AXML) *Which extension of AXML can properly address both document and workflow management? Is it mainly related to the AXML engine? Do we need light extensions of the formalism? Do we need deeper ones?*

Termination of materialization. When AXML services are invoked, materialization becomes a recursive process, since calling an AXML service may return some data that may contain new service calls that also need to be materialized, and so on recursively. Hence the following questions arise: Can we “fully” materialize a document? Can we decide in advance if the materialization of a document can be performed in a finite time?

It turns out that the termination problem is undecidable in general. There are, however, some restrictions on both the data model and the allowed services under which termination of document materialization becomes decidable. Anca Muscholl, Luc Segoufin and Serge Abiteboul have shown that materialization can be seen as a game whose objective is to provide an answer of a given type [42]; whether a winning strategy for this game yields a unique document is generally undecidable, except for some restricted class of documents [42, 6]. To achieve our overall objectives as stated in Section 1.2, there is a need to extend these results to a distributed version of the above game, for a larger class of documents. Therefore the following research problem will be considered:

Problem 2 (termination of materialization) *Study important restrictions such that is it possible to statically decide termination for AXML systems, beyond the simple cases considered so far. In particular, is it possible to statically decide termination for the AXML system? Study local materialization strategies that guarantee termination.*

It may be that such a detection is decidable in a centralized way, but not in a distributed way.

Confluence. A peer-to-peer *AXML system* involves several AXML peers communicating with each other via service calls. An important issue for such systems is that of *confluence*.

If several peers interact, the way materialization is performed may in general change the results returned. If the answer to a query is intensional, further completing the evaluation may return different data or documents, than if extensional answer was returned from the very beginning. If ultimate extensional results do not depend on the way materialization is performed, we say that the AXML system is *confluent*.

Confluence may not be always wanted. For some cases, data may change with time. For instance, the temperature in Paris may change daily, and the news headlines may change every hour. We can thus call several times the same service and obtain different answers (e.g., because the external data changed), so the same document at different times will have different semantics, thus reflecting world changes. In such cases, different choices for materialization will result in different extensional results. For other cases, however, materialization may just be a strategy for evaluation, and therefore *confluence may be critical to guarantee determinism*. This is typically the case for workflow management applications.

Positivity was proposed in [3] as a restriction that guarantees confluence. Positivity can only hold for applications without control states or updates. It is a useful and relevant case to consider for data management. However, workflow management will typically involve states and, therefore, confluence must be guaranteed by other means. Solving the following problem will be the central objective of this project:

Problem 3 (confluence) *Find conditions and means to guarantee confluence, for more general AXML systems possibly involving states.*

To achieve this, DocFlow will build on the preliminary results of the terminating ASAX project (see Section 2.2). The central effort of ASAX is on developing an *unfolding semantics* of AXML. Unfoldings are an efficient way to finely capture concurrent executions of AXML systems, at a grain where the interaction of control and tree documents can be made explicit. For stateful AXML systems, positivity does not hold at the level of documents themselves, but is re-obtained *at the level of unfoldings*, i.e., histories of documents. Our approach to the confluence problem will exploit this observation.

Guiding materialization with schemas. Schemas (such as DTD and XML Schema) are used to specify the desired format of the exchanged data in classical Web services, as part of their WSDL descriptions. They can also be used in guiding materialization of AXML documents.

As an example, consider a user requesting for a list of feasible activities in a given area, where feasibility depends e.g. on the weather forecast (see hiking). We suppose that calls to the weather forecast are charged with a fee. If the user requires to minimize the overall fees, then the system should first materialize the list of available events. The forecast service is then called only if there is some available outdoor event. Thus we use schemas to control the exchange of intensional data (e.g. the type of available events). Furthermore, schemas also

entail information about which parts of the data are allowed to be intensional and which service calls may appear in the documents, and where.

The general problem of whether a document can be rewritten to match an exchange schema turns out to be undecidable [42]. To overcome this difficulty, our algorithms consider restricted, yet realistic, cases [42, 6].

Problem 4 (guided materialization) *Extend the decidable cases for guided materialization, to distributed AXML documents.*

Monitoring. When some complex activity is performed, one would like to keep traces of what is going on. Such traces are possibly very large distributed data. One would like to query such traces. For instance, suppose one asks a query about all the factories that supply tables. After the answer is obtained, one may want to know why some particular factory answered negatively.

The most interesting aspect is the “on-line” side of the problem. For instance, one may want to obtain the name of the factories that answered negatively while the process is still computing, in particular to obtain information about its progress or lack of. This monitoring problem is important for AXML. It is important for Web services in general. Monitoring is not simply about tracing control since one also wants to trace some data that is exchanged.

One important and novel issue there is that monitoring should be performed in a distributed way by a subset of the peers involved in the AXML system, call them the *supervision peers*. Each supervision peer is in charge of one particular *domain* for supervision. Supervision peers may exchange information to achieve a consistent set of local views of what happened in their own domain, making sure that the views for the different domains are consistent across domains. This is alike the problem of distributed diagnosis that has been extensively studied by Eric Fabre and Stefan Haar, in cooperation with partners of this project [23, 10, 1, 22].

Problem 5 (monitoring) *Develop and implement techniques for distributed monitoring of AXML systems using AXML itself as an infrastructure.*

Distributed query optimization. Distributed data management has been an important domain of research almost since the early days of databases [50]. With the development of the Web and the existence of universal standards for data exchange, this problem becomes arguably a most essential challenge. Because of the heterogeneity and autonomy of sources and the scale of the Web, the problem is very complex. We believe that now is the time to develop (based on AXML) a unified algebraic framework for describing, in a very flexible manner the deployment, evaluation and optimization of queries in a collaborative manner.

By analogy with the use of relational calculus or SQL as a logical language for centralized table data, AXML can be seen as providing a *logical language for distributed tree data*. It remains to develop an *algebraic* distributed counterpart to support query optimization. One can envision an algebraic evaluation of

queries performed by collaborating query processors installed on different peers exchanging AXML data in a streaming manner. Query optimization is performed also in a distributed manner by algebraic query rewriting. One should be able to describe standard distributed query optimization techniques using the proposed framework and simple rewrite rules. This should also serve as the basis for developing novel optimization techniques.

Optimization is typically guided by heuristics and by statistics gathered about data. It is thus necessary to be able to estimate important measures at an atomic level (“cost” of communication between two sites) or at the level of complex evaluation plans (e.g., costs of evaluating a join between data from several sites).

Problem 6 (global optimization) *Develop a framework for distributed query optimization and novel techniques supporting optimization across peers, for possibly non positive AXML systems.*

Open systems, security and games. An AXML system is an open system in the sense that, from the viewpoint of a given peer, the way other peers react is not determined. Thus, when performing her actions, a peer will need a strategy allowing her to reach an objective no matter how the other peers answer. Here are examples of such situations.

The first example concerns security constraints. Suppose for instance that certain peers are known to be less secure than others. Then, we would like to guide the materialization of AXML documents in such a way that the use of services located on insecure peers is minimized.

The second example is the orchestration of several AXML peers: a global controller ensures that the combined behavior of services is restricted in such a way that it can simulate a service specification.

Another example relates to non-atomicity of service calls. Suppose that an AXML system can be simultaneously involved in materializing several documents. Since service call executions cannot be assumed atomic, different executions can interleave, which can result in undesired effects. Local controllers could guarantee that at least parts of the calls are executed in an atomic way.

Problem 7 (security) *Controller synthesis for AXML systems, seen as multi-player (or stochastic) games; the controlled systems should satisfy security properties such as atomicity or risk minimization.*

3.2 Detailed work programme

The project is structured into work packages. Each work package collects a subset of the problems listed in Section 3.1.2. For each work package, we indicate its main contributors and emphasize the *responsible(s)*.

Each work package will consist of fundamental studies, preferably consolidated by prototypes. Prototyping will aim at demonstrating feasibility. A prototype will be implemented at least in some work-packages. This includes

monitoring and optimization, and to demonstrate the use of unfoldings. Prototypes are not explicitly listed in the deliverables.

All prototypes will be connected to the AXML platform under development at the Gemo team. This team will host the developers for short periods whenever needed.

Work package 1: Convergence data/workflow, detailed specification of the problems and the model — Problem 1

This work package will involve the entire project team for the first 9 months period. Main participants will be

- *Albert Benveniste* and Stefan Haar, from the DistribCom team; they will bring their expertise in orchestrations and models of concurrency.
- *Serge Abiteboul*, from the Gemo team, Tova Milo, and Victor Vianu, foreign participants affiliated to the Gemo team; they will bring their dual expertise in AXML and orchestrations.
- PhDs: Sidney Rosario, candidate doctorant at DistribCom team, will have this as part of his research objectives.

Deliverable: a report at $t_0 + 12$. This report will in particular propose a toy but illustrative example that will serve to demonstrate the results of the project.

Work package 2: Materialization, termination, and confluence — Problems 2, 3, and 4

This is the central work package of the project, where the technology of unfoldings for AXML will be finalized and exploited for the above analysis purposes. This work package will cover the first half of the project (18 months). Main participants will be

- *Blaise Genest*, Eric Fabre, Stefan Haar, and Loic Hélouët, from the DistribCom team, will bring their expertise in unfoldings and models of concurrency;
- Anca Muscholl and Olivier Serre, from the Liafa team, will bring their expertise in games and decidability studies for distributed systems;
- Serge Abiteboul, from the Gemo team, will have a moderate participation in this work package; his role will consist in checking that the unfolding semantics proposed for AXML conforms the intuitive semantics.
- PhDs: Sidney Rosario, candidate doctorant at DistribCom team, will contribute to developing the unfolding semantics.

Deliverables:

- a report on unfoldings for AXML at $t_0 + 12$.
- a summarizing report at $t_0 + 24$.

Work package 3: Monitoring — Problem 5

This work package will address the monitoring of AXML systems as described in Problem 5. This will reuse the unfolding technology developed in Work package 2 and the background from DistribCom team as well as the background on datalog from Gemo team. This work package will start at month $t_0 + 9$, its main participants will be:

- *Eric Fabre*, Albert Benveniste, Stefan Haar and Loic H elou et, from the DistribCom team, will bring their expertise in unfoldings and diagnosis;
- Serge Abiteboul from the Gemo team will bring his expertise in Web monitoring and continuous services.
- PhDs: Bogdan Marinoiu, currently PhD student in the Gemo team, will work on monitoring. One PhD from the DistribCom team will work part time on this subject (the other part will be devoted to optimization across peers), in relation with QoS issues.

Deliverables:

- a report at $t_0 + 24$.
- a report at $t_0 + 36$.

Work package 4: Optimization across peers — Problem 6

This work package will consist in developing a formal framework to capture distributed data management, by extending and refining AXML. This should in particular capture data and service replication/distribution and migration. It should be based on statistical and other metrics to assess performance. Novel heuristics and strategies for optimization should be proposed. This work package will start at month $t_0 + 6$, its main participants will be:

- *Serge Abiteboul* and Ioana Manolescu, from the Gemo team will bring their expertise on (distributed) query optimization.
- Eric Fabre and Albert Benveniste, from the DistribCom team, will bring their expertise in statistics and distributed optimization.
- PhDs: Nicoleta Preda, PhD student in the Gemo team, will work part time on the topic of on optimization; one more doctorant is planed at Gemo team full time on this subject. One PhD student from the DistribCom team will work part time on this subject (the other part will be devoted to monitoring), in relation with QoS issues.

Deliverables:

- a report at $t_0 + 24$.
- a report at $t_0 + 36$.

Work package 5: Security — Problem 7

Expected results of this work package is are: 1/ game models for several security issues such as atomicity and correct materializations in the AXML framework, 2/ game models for confidentiality in web services, and 3/ a controller synthesis framework based on the game models listed above to enhance security in AXML systems. This work package will start at month $t_0 + 6$, its main participants will be:

- Loic H elou et, from the DistribCom team, will bring his expertise in models of concurrency and experience from studying cover channels;
- *Anca Muscholl* and Olivier Serre, from the Liafa team, will bring their expertise in games and models of concurrency.
- Victor Vianu, affiliated to the Gemo team, will bring his expertise in distributed Web services verification.
- PhDs: Debmalya Biswas, PhD student in the DistribCom team, will work on security and atomicity of AXML transactions.

Deliverables:

- a report at $t_0 + 24$.
- a report at $t_0 + 36$.

4 Deliverables and responsibilities

Deliverables are synchronized with long reports and are therefore delivered on an annual basis. The deliverables and responsibilities are shown in Table 1.

	responsible	team	$t_0 + 12$	$t_0 + 24$	$t_0 + 36$
WP 1	A. Benveniste S. Abiteboul	DistribCom Gemo	$R_{1,\text{final}}$		
WP 2	B. Genest	DistribCom	$R_{2,1}$	$R_{2,\text{final}}$	
WP 3	E. Fabre	DistribCom		$R_{3,1}$	$R_{3,\text{final}}$
WP 4	S. Abiteboul	Gemo		$R_{4,1}$	$R_{4,\text{final}}$
WP 5	A. Muscholl	Liafa		$R_{5,1}$	$R_{5,\text{final}}$

Table 1: Deliverables and responsibilities.

5 Detailed expected results and criteria for success

Problem 1: convergence workflow/AXML. This topic is somehow open ended. Still, we expect to be able, by the mid of the project, to deeply understand the intertwining between control and data in peer-to-peer data and workflow management—this is our criterion for success. Having achieved this should help us getting better understanding of AXML from this point of view, and let us possibly propose upgrades or evolutions for AXML, its language and/or the way queries are answered. A byproduct of this work package will be a toy but illustrative example that will serve to demonstrate our results throughout the project.

Problems 2, 3, 4: confluence and materialization. By the mid of the project, we expect to have shown how unfolding techniques can contribute to solving the confluence problem. This will also contribute to refining the class of queries that should be considered, and the way such queries should be answered—these are our criteria for success. At the same time and possibly also later in the project, fundamental results on decidability for the termination of materialization should be obtained. Having this at hand, we plan to demonstrate our approach by formally verifying those properties on a toy but demonstrative example.

Problem 5: distributed monitoring of AXML systems. By the end of the project, AXML will be enhanced with a framework allowing for the distributed monitoring of AXML systems, on simple questions such as the flow of control followed by a query, and/or the types of data exchanged—this is our criterion for success. This will be demonstrated on our above mentioned toy but illustrative example. We shall also perform a preliminary investigation of the monitoring of more hidden properties that may not be directly visible from the AXML peers, but are correlated to what the peers can observe.

Problem 6: global optimization. By the end of the project, we plan to demonstrate on our above mentioned toy but illustrative example, a framework for distributed query optimization and novel techniques supporting optimization across peers. This framework will aim at optimizing the latency of queries by selecting the answering peer among functionally equivalent ones—this is our criterion for success. We will propose this framework for use with AXML. This will open the way to studies on more sophisticated problems related to end-to-end performance and quality of service.

Problem 7: security. Security is a very wide research area. We have indicated in the project description ways of addressing this area (in particular games). We will, however, spend a first phase of the project in properly identi-

fyng those security problems that are specific to AXML and thus justify specific research.

6 The teams and their qualification

For each team, key persons of the group are listed next, together with their involvement in the project. Then, we comment on cross-disciplinarity and indicate international collaborations.

6.1 The Liafa team and its key persons

Anca Muscholl⁵ (40%). Anca will head the project. She received her PhD in Computer Science from the University of Stuttgart in 1994, and her habilitation degree in 1999. She is a professor for Computer Science at the University Paris 7 and member of the Liafa since 1999. Her current research interests include: game theory, automata, logics, XML, automated verification, design and validation of concurrent systems, models for asynchronous communicating systems. She worked on algebraic and logical foundations of concurrent systems, with emphasis on automata-based model checking methods. Recent contributions focused on models for communicating systems, in particular on message sequence charts, and on model checking communicating finite state systems. Very recently she worked on automata and logics for XML-based data and on game-related aspects of AXML. She has published about 45 conference and journal papers. She is involved in the EU-TMR project GAMES, was a member in programme committees of conferences such as STACS, CONCUR, CAV, TACAS, FOSSACS etc., and she is field editor of the electronic journal DMTCS. She organizes the EPIT Spring School 2006 Games in Semantics and Verification.

Olivier Serre (20%). Olivier graduated from Ecole Nationale Supérieure de Cachan in 1999. Ms.C. in computer science and maths, and PhD in computer science in 2001, ENS Cachan, and 2004, Paris VII, respectively. He was post-doctoral researcher at University of Aachen in 2004-2005. Since then, he is with CNRS, Liafa, Paris. His research interests include temporal logics and games.

6.2 The Gemo team and its key persons

Serge Abiteboul⁶ (30%). Serge Abiteboul received his Ph.D. in Computer Science from the University of Southern California in 1982, and his French Thèse d'Etat from the University of Paris XI in 1986. He is Senior Researcher at INRIA and manages the Gemo Database Group. He has held professor positions at Stanford U. and the Ecole Polytechnique. He is one of the co-authors of *Foundations of Databases*, the reference in database theory, and of

⁵www.liafa.jusieu.fr/~anca

⁶www-rocq.inria.fr/~abitebou

Data on the Web. He has worked recently on object databases, digital libraries, semistructured data, data integration, electronic commerce, web systems. He received the 1998 ACM SIGMOD Innovation Award. He has been program chair of a number of conferences including ACM PODS-1995, ICALP-1994, ICDT-90, ECDL-99 and the EDBT99 Summer School, and VLDB-03. He has been chair of the executive committee ACM SIGACT-SOGMOD-SIGART PODS. He co-founded in 2000 a start-up, named Xyleme, where he is now Scientific Advisor.

Ioana Manolescu ⁷ (20%). Ioana obtained her PhD in 2001, from the University of Versailles. After a post-doc at Politecnico di Milano, she joined the Gemo group of INRIA-Saclay, France, where she now is researcher. Ioana has worked on models and systems for large and distributed data management, and on XML query and update optimization. Since 2005, Ioana is also a member of the W3C XQuery Working Group Task Force for and update-oriented extension of XQuery. Ioana has been involved in two European and six French-funded R&D projects. In particular, she currently is the co-recipient of a "Young Researcher" French grant, and scientific manager of the "Peer-to-peer" work package in a recent French R&D project on Web surveillance.

Victor Vianu ⁸ (15%). Victor Vianu is professor at UC San Diego, where he is heading the data base group. Victor spends each year several weeks with the Gemo team. This will allow for an efficient and streamlining cooperation between the two teams of Gemo and UC San Diego. Victor and Serge have co-authored the classics *Foundations of data bases*, with R. Hull. Victor is area editor for database systems and theory at JACM.

6.3 The DistribCom team and its key persons

The DistribCom⁹ team of INRIA performs algorithmic studies supporting distributed management of telecommunications networks/services and Web services. The group is widely recognized in in area of models of concurrency (event structures and nets), both functional and for QoS and probabilistic aspects. It has developed techniques for the distributed monitoring of telecommunications networks/services. More recently, the group entered the field of Web services, where it focuses on semantics and QoS issues. The group has long term cooperations with Alcatel and France Telecom R&D. Since 2005, the group has a cooperation with the two other partners of DocFlow, on the AXML formalism. Key persons of the group are listed next, together with their involvement in the project:

⁷<http://www-rocq.inria.fr/~manolesc/>

⁸<http://www-cse.ucsd.edu/~vianu/>

⁹<http://www.irisa.fr/DistribCom/>

Albert Benveniste¹⁰ (30%). Albert Benveniste performed his These d'Etat in Mathematics, probability theory, in 1975. From 1976 to 1979 he was associate professor in mathematics at Universite de Rennes I. Since 1979 he has been Directeur de Recherche at INRIA-Rennes. His current interests include: system identification and change detection in signal processing and automatic control, vibration mechanics, reactive and real-time embedded systems design in computer science, and network and service management in telecommunications. He has coauthored with M. Metivier and P. Priouret the book *Adaptive Algorithms and Stochastic Approximations*. He has been co-inventor, jointly with Paul Le Guernic, of the synchronous language *Signal* for embedded systems design. Since 1996, he has been active in distributed algorithms for network and service management in telecommunications, where he has contributed to distributed fault diagnosis. Albert Benveniste is or has been Associate Editor for several journals in Control and PC member of many conferences in computer science. In 1980 he was winner of the IEEE Transactions on Automatic Control Best Paper Award for his paper on blind deconvolution in data communications. In 1990 he received the CNRS silver medal and in 1991 he has been elected IEEE fellow.

Eric Fabre (30%). Eric graduated from the Ecole Nationale Supérieure des Télécommunications, Paris. He received the M.Sc. degree in probability and statistics and the PhD in electrical engineering from the university of Rennes I, in 1990, 1993, and 1994, respectively. In 1995, he was postdoctoral researcher at LADSEB-CNR, Padova, Italy. Since then, he has been with INRIA, Rennes. His research interests include Bayesian networks, distributed discrete event systems, and their applications to telecommunications network and service management.

Blaise Genest¹¹ (25%). Blaise graduated in CS in 2001 from Ecole normale supérieure de Cachan. He obtained his PhD from the University Paris VII in 2004. He was with the group of Doron Peled at the University of Warwick, UK, and is now Chargé de Recherche at CNRS. His research interests include concurrent systems, temporal logics, and model checking.

Stefan Haar¹² (30%). Stefan Haar studied mathematics at Hamburg University, Germany, and Johns Hopkins University, Baltimore, MD/USA. He received the *Diplom* (M. Sc.) degree in mathematics (stochastic bifurcation theory) and the PhD degree in computer science (Petri Net theory) both from Hamburg University, in 1992 and 1997. As a PostDoctoral Researcher, he took part in the PEPtool project at Humboldt University, Berlin/Germany, and in the EU project ALAPEDES (algebraic approach to performance evaluation of discrete event systems), at INRIA Nancy and ENS Paris, France. Since 2001, he has been with INRIA Rennes as Chargé de Recherche. His research interests include

¹⁰<http://www.irisa.fr/DistribCom/benveniste/>

¹¹<http://perso.crans.org/~genest/>

¹²http://www.irisa.fr/DistribCom/Personal_Pages/shaar/index.html

distributed systems, partial order semantics, timed and probabilistic systems, fault diagnosis, telecommunication systems, and web services. Stefan Haar is currently directing the RNRT-funded research project SWAN (Self-Aware Management) and the research action (ARC) ASAX on analysis of Active XML systems. He is an Associate Editor of IEEE Transactions on Automatic Control.

Loïc Hérouët¹³ (25%). Loïc Hérouët studied computer science at the university of Rennes 1. After graduating from INSA de Rennes in 1996, he received a PhD in computer science from University of Rennes 1 in 2000. After his PhD, he spent one post doctoral year in France Telecom and took in the IST project INTERVAL. Since 2001, he is chargé de recherche at INRIA Rennes. His current research interests are distributed systems, partial order models, game theory, diagnosis, and security. Loïc Hérouët takes part in the RNRT Persiform, and in the research action ASAX. He also takes part in the INRIA-ARC Potestat, which aims at testing security policies.

6.4 Complementarity of the teams and cross-disciplinarity of the consortium

The consortium mixes teams with very different backgrounds:

- The Gemo team has its roots in the area of data bases.
- The Liafa team has its background in formal models and methods for verification and games.
- The DistribCom team is a mix of computer scientists with background in concurrency theory and control scientists with background in applied maths and statistics.

These teams already have a beginning cooperation together in the framework of the ASAX cooperative research action of INRIA.

6.5 International collaboration

Tova Milo¹⁴ is professor at Tel Aviv University. She is a specialist in data management and one of the original authors of Active XML. She recently worked on Web services. Tova will keep contact with the Gemo team on topics related to this proposal.

William Cook¹⁵ and Jadayev Misra¹⁶, from the Department of Computer Sciences, University of Texas at Austin, will keep in contact with the DistribCom team on the topics of this project. Jay is well known for his contributions in the area of distributed systems and the Unity language; he is IEEE and ACM Fellow.

¹³<http://www.irisa.fr/DistribCom/PersonalPages/helouet/LHengpage.html>

¹⁴<http://www.math.tau.ac.il/~milo/>

¹⁵<http://www.cs.utexas.edu/users/wcook/>

¹⁶<http://www.cs.utexas.edu/users/misra/>

He and Willam have developed a clean and elegant formalism for Web services orchestrations, called ORC¹⁷. ORC is currently experimented at DistribCom for studies on end-to-end QoS.

References

- [1] S. Abiteboul, Z. Abrams, S. Haar, and T. Milo. Diagnosis of Asynchronous Discrete Event Systems - Datalog to the Rescue! In *Proc. of ACM PODS*, 2005.
- [2] S. Abiteboul, B. Amann, S. Cluet, A. Eyal, L. Mignet, and T. Milo. Active Views for Electronic Commerce. In *Proc. of VLDB*, 1999.
- [3] S. Abiteboul, O. Benjelloun, and T. Milo. Positive Active XML. In *Proc. of ACM PODS*, 2004.
- [4] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web: From Relations to Semistructured Data and XML*. Morgan Kaufmann Publishers, San Francisco, California, 1999.
- [5] S. Abiteboul, I. Manolescu, and E. Taropa. A framework for distributed data management. In *Proc. of Intern. Conf. on Extending Database Technology*, 2006.
- [6] S. Abiteboul, T. Milo, and O. Benjelloun. Regular and Unambiguous Rewritings for Active XML. In *Proc. of ACM PODS*, 2005.
- [7] A. Acharya, M. Ranganathan, and J. Saltz. Sumatra: A language for resource-aware mobile programs. In *Mobile Object Systems: Towards the Programmable Internet*, number 1222, pages 111–130. Springer, LNCS, 1997.
- [8] The Active XML homepage.
<http://activexml.net/>.
- [9] P. Baroni, G. Lamperti, P. Pogliano, and M. Zanella. Diagnosis of large active systems. *Artificial Intell.*, 1999.
- [10] A. Benveniste, S. Haar, E. Fabre, and C. Jard. Fault diagnosis for distributed asynchronous dynamically reconfigured discrete event systems. In *Proc. of the 16th IFAC World Congress*, Prague, july 2005.
- [11] The BitTorrent Homepage.
<http://www.bittorrent.com>.
- [12] R. Boel and J. van Schuppen. Decentralized failure diagnosis for discrete event systems with costly communication between diagnosers. In *WODES'02*.
- [13] A. Bonifati, D. Braga, A. Campi, and S. Ceri. Active XQuery. In *Proc. of ICDE*, 2002.
- [14] A. Bonifati, S. Ceri, and S. Paraboschi. Pushing Reactive Services to XML Repositories using Active Rules. In *Proc. of the Int. WWW Conf.*, Hong Kong, China, May 2001.
- [15] Business Process Execution Language for Web Services Version 1.1. Available from
<http://www.ibm.com/developerworks/library/ws-bpel/>.
- [16] Macromedia Coldfusion MX, 2004.
<http://www.macromedia.com/software/coldfusion/>.

¹⁷<http://www.cs.utexas.edu/users/wcook/projects/orc/>

- [17] O. Contant and S. Lafortune. Diagnosis of modular discrete event systems. In *WODES'04*.
- [18] In G. Weikum, editor, *Infrastructure for Advanced E-Services*, volume 24, no. 1. Bulletin of the Technical Committee on Data Engineering, IEEE Computer Society edition, Mar 2001.
- [19] A. Deutsch, M. Marcus, L. Sui, V. Vianu, and D. Zhou. A verifier for interactive, data-driven web applications. In *SIGMOD Conference*, 2005.
- [20] A. Deutsch, L. Sui, and V. Vianu. Specification and verification of data-driven web services. In *PODS*, pages 71–82, 2004.
- [21] A. Deutsch, L. Sui, V. Vianu, and D. Zhou. A verifier for interactive, data-driven web applications. In *PODS Conference*, 2006. to appear.
- [22] E. Fabre. Distributed Diagnosis based on Trellis Processes. In *44th Conf. on Decision and Control (CDC)*, Sevilla, Spain, dec 2005.
- [23] E. Fabre, A. Benveniste, S. Haar, and C. Jard. Distributed monitoring of concurrent and asynchronous systems. *Discrete Event Dynamic Systems*, 15(1):33–84, Mar 2005.
- [24] L. Galanis, Y. Wang, S. Jeffery, and D. DeWitt. Locating data sources in large distributed systems. In *VLDB*, 2003.
- [25] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. Ullman, and J. Widom. The TSIMMIS Approach to Mediation: Data Models and Languages. *Journal of Intelligent Information Systems*, 8:117–132, 1997.
- [26] S. Genc and S. Lafortune. Distributed diagnosis of discrete-event systems using petri nets. In *24th Int. Conf. on Applications and Theory of Petri Nets*.
- [27] The Gnutella homepage.
<http://www.gnutella.com>.
- [28] A. Gupta. *Integration of Information Systems: Bridging Heterogeneous Databases*. IEEE Press, 1989.
- [29] H. Gupta. Selection of Views to Materialize in a Data Warehouse. In *Proc. of ICDT*, pages 98–112, 1997.
- [30] L. Haas, D. Kossmann, E. Wimmers, and J. Yang. Optimizing queries across diverse data sources. In *VLDB*, 1997.
- [31] A. Halevy, Z. Ives, P. Mork, and I. Tatarinov. Piazza: data management infrastructure for semantic web applications. In *WWW*, 2003.
- [32] J. M. Hugh, S. Abiteboul, R. Goldman, D. Quass, and J. Widom. Lore: A Database Management System for Semistructured Data. Technical report, Stanford University Database Group, Feb 1997.
- [33] R. Hull, M. Benedikt, V. Christophides, and J. Su. E-services: a look behind the curtain. In *Proc. of ACM PODS*, 2003.
- [34] Jelly: Executable XML.
<http://jakarta.apache.org/commons/sandbox/jelly>.
- [35] The Join-Calculus.
<http://join.inria.fr/>.
- [36] The Kazaa Homepage.
<http://www.kazaa.com>.

- [37] O. Kupferman and M. Vardi. Synthesizing distributed systems. In *16th Annual IEEE Symposium on Logic in Computer Science, LICS'01*, pages 389–398. IEEE, 2001.
- [38] A. Levy, A. Rajaraman, and J. Ordille. Querying Heterogeneous Information Sources Using Source Descriptions. In *Proc. of VLDB*, pages 251–262, 1996.
- [39] P. Madhusudan and P. S. Thiagarajan. Distributed controller synthesis for local specifications. In *Automata, Languages and Programming, 28th International Colloquium, ICALP'01*, number 2076 in LNCS, pages 396–407. SPRINGER, 2001.
- [40] I. Manolescu, D. Florescu, and D. Kossmann. Answering XML queries over heterogeneous data sources. In *Proc. of VLDB*, 2001.
- [41] The Morpheus homepage.
<http://www.morpheus-os.com>.
- [42] A. Muscholl, T. Schwentick, and L. Segoufin. Active Context-Free Games. In *Proc. of STACS*, 2004.
- [43] N. Kavantzias and D. Burdett and G. Ritzinger and T. Fletcher and Y. Lafon. Web Services Choreography Description Language – WS-CDL, version 1.0.
<http://www.w3.org/2002/ws/chor/edcopies/cdl/cdl.html> .
- [44] The Napster homepage.
<http://www.napster.com>.
- [45] A. Pnueli and R. Rosner. Distributed reactive systems are hard to synthesize. In *13th IEEE Symp. FOCS*, pages 746–757. IEEE, 1990.
- [46] J. Powell and T. Maxwell. Integrating Office XP Smart Tags with the Microsoft .NET Platform.
<http://msdn.microsoft.com>, 2001.
- [47] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proc. IFIP/ACM Middleware 2001.*, Heidelberg, Germany, 2001.
- [48] Simple Object Access Protocol (SOAP) 1.1.
<http://www.w3.org/TR/SOAP>.
- [49] R. Su, W. Wonham, J. Kurien, and X. Koutsoukos. Distributed diagnosis for qualitative systems. In *WODES'02*.
- [50] T. Özsu and P. Valduriez. *Principles of Distributed Database Systems, 2nd Edition*. Prentice-Hall, 1999.
- [51] J. Ullman. *Principles of Database and Knowledge Base Systems*. Computer Science Press, 1989.
- [52] V. Vassalos and Y. Papakonstantinou. Describing and using the query capabilities of heterogeneous sources. In *VLDB*, 1997.
- [53] J. Widom and S. Ceri. *Active Database Systems: Triggers and Rules for Advanced Database Processing*. Morgan Kaufmann Publishers, 1996.
- [54] G. Wiederhold. Intelligent Integration of Information. In *Proc. of ACM SIGMOD*, pages 434–437, Washington, DC, May 1993.
- [55] Web Services Definition Language (WSDL).
<http://www.w3.org/TR/wsdl>.

- [56] Web Services Flow Language (WSFL 1.0).
Available from <http://www.ibm.com/>.
- [57] XLANG, Web Services for Business Process Design.
http://www.gotdotnet.com/team/xml_wsspecs/xlang-c.
- [58] The Extensible Markup Language (XML) 1.0 (2nd Edition).
<http://www.w3.org/TR/REC-xml>.
- [59] The XML Schema specification.
<http://www.w3.org/TR/XML/Schema>.
- [60] XQuery 1.0: An XML Query Language.
<http://www.w3.org/TR/xquery>.
- [61] T. Yoo and S. Lafortune. A general architecture for decentralized supervisory control of discrete-event systems. *Discrete Event Dynamic Systems: Theory and Applications*, 2002.