From Static to Dynamic Environments: *CON*sistency, Recovery and *DEP*endability Issues (CONDEP)  
TIN2006-14738-C02

Francisco Daniel Muñoz Escoí *  
Instituto Tecnológico de Informática, Univ. Politécnica de Valencia  
José Ramón González de Mendivil Moreno †  
Dept. de Ing. Matemática e Informática, Univ. Pública de Navarra

**Abstract**

Nowadays, there are good research results in the fields of dependability and data consistency in distributed static environments, as proven by recent publications on secure and replicated databases. However, even in those environments there exist some issues that can be improved, such as their recovery protocols and the design of frameworks for maintaining concurrently multiple replication protocols specially tailored for the current set of applications that access a given replicated database, or for exchanging at runtime those protocols. Besides this, new and improved mobile devices exist, and wireless networks may be found and used anywhere, so their users will ask soon for dependable collaborative applications. Thus, the distributed protocols run by these mobile devices will be one of the key issues for improving their dependability, being distributed agreement the most important. This project is focused on: (1) the improvements described above for static environments; (2) the design, specification and implementation of a dependable architecture for dynamic environments; (3) new supports for group membership, distributed agreement, secure communications, authentication, and consistency and replication management in those dynamic environments.  

**Keywords:** Dependability, Replication Protocols, Recovery Protocols, Dynamic Distributed Systems.

1 **Project Objectives**

In the following list, the specific goals of this project are detailed:

1. **Definition of basic support and design for distributed agreement in dynamic systems.** For static systems, there exist results about the support needed to achieve agreement that are widely accepted by the scientific community. Many of these results

---

*Email: fmunyoz@iti.upv.es*  
†Email: mendivil@unavarra.es
can not be applied to dynamic systems due to huge differences between these systems. Therefore, with this target, we intend to define a starting point from which we will build a software that needs distributed agreement in dynamic systems. To do so, we first need to revise the consensus concept itself. Once the current proposals have been studied, the basic mechanisms for consensus in dynamic environments must be designed.

2. **Reliability improvements in dynamic systems starting from distributed agreement.** Most existing replication proposals will need significant adaptations to be usable in ad-hoc networks, especially when distributed agreement basic algorithms have undergone significant modifications.

3. **Component development to improve dynamic systems reliability.** To the extent of results delivered by targets 1 and 2, we will implement components to improve dynamic systems reliability. In this project we intend to execute, measure and compare the different protocol alternatives that we design.

4. **Improvement of a data replication middleware.** In order to carry out this improvement, we shall accurately investigate some support mechanisms that can be implemented in the underlying DBMS itself, using standard mechanisms. Experience acquired in our previous projects (where our MADIS middleware was developed) demonstrates that this approach can vastly improve the middleware performance. Moreover, we also intend to support partial replication, which will affect fundamentally the goals 5, 6 and 8. When an application must manage a huge amount of data, it makes sense to use partial replication in order to minimise storage consumption in each node.

5. **Development of flexible replication protocols.** For data replication we shall develop a schema that permits the simultaneous use of multiple protocols (i.e., providing support to multiple applications) in the same database system. This will also enable dynamic swapping of protocols. At last, we will develop new protocols for dynamic systems.

6. **Development of efficient recovery/reconciliation protocols.** Study, design and analysis of different recovery strategies for different replication protocols that already exist or will be obtained in target 5, either for static or dynamic systems. Performance analysis.

7. **Use of techniques based on fuzzy logic.** We will study which effects these techniques may have on improving the envisaged support in targets 1, 5 and 6, as detailed above.

8. **Development of support for integrity constraints evaluation in distributed systems.** The internal middleware design will be extended in order to enable integrity evaluations in replicated databases. Also the more complex case of partial replication will be approached.

In order to achieve these goals, multiple concrete tasks were specified in the project proposal and a basic schedule was designed. Table 1 summarises them.

## 2 Level of Achievement

Let us explain which is the current level of achievement in each one of the objectives and tasks described in the previous section. To do so, we use a numbered list that matches the
Table 1: Project Tasks and Schedule.

The objective numbers presented in Section 1. Task numbers were also shown in Table 1. In general, all planned tasks have been able to provide some relevant results that have been already published in specialised symposia or conferences. So, all the objectives have currently been (partially) achieved as expected in our initial schedule.

1. Definition of basic support and design for distributed agreement in dynamic systems. Once we had reviewed the existing works on dynamic distributed agreement, as suggested by task 1.1, the approach followed in order to manage such agreement procedures is based on stability criteria. Our aim with such criteria is to solve agreement in a stable subset of the nodes that compose the intended distributed system (i.e., a subset that shares most characteristics of a static distributed system, and that can use traditional agreement algorithms), sharing later the agreed value with other more dynamic nodes, as soon as they interact with such stable core system. Such criteria depend a lot on the application or problem that raised the agreement needs. Thus, a possible application in a dynamic system would be the membership protocols being used. In such case, we have proposed [30] four different kinds of stability criteria that are based on: time, distance, node energy level, and application-specific parameters. Such paper can also be considered a result from task 1.4 (Design of distributed agreement protocols).

Considering task 1.2 (Improvement of the membership estimation service), only the proactive OLSR routing protocol was able to provide some membership estimation information. So, our aim in such task is to enhance such kind of support in MANETs. To this end, we took as our basis the reactive DSR protocol and complemented it with a gossip-style membership management. The resulting combination (our GDSR protocol) has been compared to OLSR in [28], where we have carefully analysed which is the best approach for each
possible environment. Such paper proves that—with an appropriate tuning—our proposed solution is able to improve the results of the original one, reducing its energy consumption and providing more accurate membership lists.

Note also that both papers [28, 30] have needed a careful selection of metrics in order to evaluate the resulting protocols. Such metrics have been generated by task 1.3.

2. **Reliability improvements in dynamic systems starting from distributed agreement.** As already outlined in the previous objective, agreement is needed as an important building block in other components of a reliable distributed system. To begin with, the membership services need agreement (task 2.2) in order to decide which are the current nodes of a reliable system. For reliable dynamic systems, such membership services are commonly based on membership estimators. We have proposed several enhancements—based on our GSDR protocol discussed above—to this kind of services that have been presented in different publications [28, 29] where different mobility and nearness patterns were combined showing that our proposal behaves better than the traditional techniques in most of the surveyed configurations.

Regarding replication protocols (task 2.3) for dynamic systems, two different research lines have been taken. On one hand, we have surveyed the recovery/reconciliation protocols needed in this kind of environments, proposing a reconciliation protocol [8] for partitionable object-oriented systems. In such protocol, system consistency was based on constraints, and each system subset/partition was able to proceed if the invocations being received from its clients do not violate the existing constraints. In the reconciliation phase, all temporary accepted operations are merged and a new full execution order is agreed for them by a single reconciliation manager process, although some of the previously accepted operations could be rejected in this procedure. Moreover, new operations are even accepted during this reconciliation procedure until the full order has been decided. At that time, system activity is temporarily paused until the agreed state is propagated to all system nodes. Anyway, since that reconciliation protocol was designed for a passive replication protocol, such pausing interval is negligible since the new selected primary is able to accept and process all new requests, holding the updates in buffers that are not transmitted to other replicas until the pause interval terminates.

On the other hand, the second line of replication protocols for dynamic systems consists in using commutative operations. Thus, no special reconciliation protocols will be needed. Besides removing the need of a reconciliation protocol, when objects have a complete interface of commutative operations another advantage can be found: operation updates do not need total-order propagation. As a result, unordered reliable multicast is enough in that case. All these issues—and their performance advantages—were presented and experimentally evaluated in [10].

3. **Component development to improve dynamic systems reliability.** The analysis of existing frameworks for developing dynamic system protocols (target of task 3.1) led us to the selection of the NS2 simulator. Some papers [28, 29] discussed in the previous objective are also samples of the results planned for task 3.3 (Development of agreement protocols). Regarding task 3.2 (Implementation of membership-routing cross-layer protocols), we have implemented and extended the evaluation of our cross-layer solution [31]. Results from
the other two tasks (3.4 and 3.5) will be described in objectives 5 and 6 in order to avoid repetition.

4. **Improvement of a data replication middleware.** Our MADIS middleware has been greatly enhanced in this first part of the project. To begin with, its DB-layer has been rewritten (this was the aim of task 4.1) and its support for writerset collection and application demands now 10% of the time needed in its previous release. This has been documented in although it has not yet been published. Regarding task 4.2 (Migration of the MADIS middleware), we have completed the middleware migration to MySQL (the original release was designed and implemented in PostgreSQL) and we plan to migrate it to Oracle in the future.

Besides rewriting its bottom DB-layer in task 4.1, the whole middleware was redesigned, redistributing its functionality through its components and extending some of their interfaces providing thus support for simplified recovery (task 6.2), partial replication (task 4.3) and to multiple isolation levels in a single replication protocol or multiple concurrent protocols in the same nodes (task 4.4). To this end, we took as a base for such redesign the requirements outlined in the technical report versions of (recovery), (partial replication), (support for multiple isolation levels in a single protocol) and (support for multiple concurrent replication protocols in the middleware, and to their dynamic exchange).

5. **Development of flexible replication protocols.** Based on the results of task 4.4, task 5.1 is centred on the design and implementation of a middleware component that allows the usage of multiple concurrent replication protocols at a time. This permits the dynamic migration from a given protocol to another one, without the need of stopping system activity, since each client process may decide which is the best protocol for its tasks. Regarding database replication protocols, such a component was designed in as stated above, and its implementation has been terminated in 2008 for protocols based on atomic broadcast. We have already done a performance evaluation of this metaprotocol, combining active, certification-based and weak voting pluggable protocols. The results have been summarised in . They confirm that our support is very flexible (protocols need only a minimal adaptation in order to be plugged into our metaprotocol, requiring less than 10 hours of implementation effort) and that introduces no overhead (transaction completion times are not increased in our metaprotocol when they are compared with their protocol stand-alone versions).

Task 5.2 is devoted to the implementation of protocols pluggable into the metaprotocol described in the previous paragraph (and such kind of protocols are also described in ), and to the development of new stand-alone protocols. For such stand-alone protocols we have followed three different trends. Firstly, we proposed in the rules to be followed in order to support multiple isolation levels in a single protocol. Such rules were extended and refined in , comparing our approach to the principles outlined in Atul Adya’s PhD thesis; , where the rules needed for supporting the Read Committed isolation level were given; and , where Adya’s mixed serialisation graphs were extended in order to accept SI transactions. This theoretical basis was used for designing and implementing several multi-level replication protocols. The first one was where the serialisable level and different variants of the SI level where supported, combining a weak voting strategy for the first isolation level and a certification-based one for the latter. Later, the SIRC protocol
was also designed and implemented. Such protocol supports the SI and RC levels (those provided by MADIS’ underlying PostgreSQL DBMS). The results from another protocol of this kind were published in the NOTERE’07 conference. Its main difference consists in using a single weak voting strategy for all supported isolation levels. Finally, the k-bound and SIRC protocols were merged into a new one named $g$-Bound. The results provided in all such papers show that multi-level protocols are able to reduce the abortion rates provided by traditional single-level protocols, since in the former transactions can select more relaxed isolation levels than in the latter, avoiding the need of abortion in some kinds of inter-transactional conflicts.

Our second trend in the replication protocols field consisted in reviewing the principles being followed in the protocols that provide the Snapshot Isolation (SI) level. We have proposed different enhancements to the regular protocols supporting such level, writing multiple technical reports on this subject. All of them were submitted to different journals. At the moment, two of them have been accepted and the other are still in different review steps.

The last trend has generated different new data replication protocols for mobile or partitionable environments (the latter in collaboration with Prof. Karl M. Göscha’s group), turn-based protocols that are able to guarantee that once a transaction multicasts its writesets, it will never get aborted (none of the previous data replication approaches was able to provide such guarantee), and variants of passive replication specially tailored for dynamic environments.

Regarding task 5.3 (Development of partial replication protocols), two of our papers have provided a good basis for developing such partial replication protocols. In the first one a protocol of this kind was designed and discussed, whilst in the second (developed in collaboration with Prof. Fernando Pedone’s group) we have analysed under which conditions the SI level will provide better results than the serialisable one in such kind of replication. Since SI is based on multi-versioned concurrency control and its results depend a lot on the number of available item versions, SI is not always better (in terms of abortion rates) than serialisable, despite being a bit more relaxed.

Results from tasks 5.4 and 5.5 (Consistency models) can be found in all papers cited in this objective. Besides them, we have also proposed an enhanced protocol in order to ensure linearisable semantics in database replication protocols for static environments, collaborating with Prof. Luís Rodrigues’ group.

6. Development of efficient recovery/reconciliation protocols. As already discussed in the results related to objective 2, we have proposed new and efficient reconciliation strategies for dynamic environments, collaborating with Prof. Simin Nadjm-Tehrani’s group. Note also that all middleware extensions needed in task 6.2 have already been described above (in objective 4).

Regarding task 6.1 (Review of existing recovery protocols), we surveyed the existing recovery solutions for both static and dynamic environments, generating several papers of this kind where the most important characteristics of each recovery strategy were described and compared.

Considering task 6.3 (Design, implementation and testing of recovery protocols), we de-
signed \cite{45,47} (in collaboration with Prof. Bettina Kemme’s group), implemented and evaluated \cite{50} multiple recovery optimisations for certification-based data replication protocols; adapting some of them to weak voting protocols \cite{33}. To our knowledge, the results shown in \cite{45,50} were the first publications providing a performance evaluation of any recovery protocol for replicated databases. Other groups have been able to publish results of this kind in 2008, but not before. Until then, recovery papers only provided the needed algorithms and their correctness proofs but not any protocol benchmarking.

In the context of task 6.3 we have also reviewed the usage of recoverable models in group communication systems and their impact on the database recovery protocols. This has generated the PhD thesis of Rubén de Juan \cite{17}. In the context of this project, such work started with an analysis \cite{20} of the logging needs of the replication strategies in a recoverable model, that was later applied to review existing database recovery protocols \cite{19}, and to enhance our own ones \cite{23}. Later, we enhanced the existing progress conditions for primary-partition environments \cite{21} and extended the virtual synchrony execution model with logging \cite{12}. The overhead introduced by this message logging technique \cite{22} has been also evaluated and has been already accepted for publication in FINA’09.

7. **Use of fuzzy logic techniques.** Results obtained in this research line are still negligible when compared to those got in the implementation of our replication protocols.

8. **Development of support for integrity constraint evaluation in distributed systems.** Task 8.1 (Review of existing integrity checking mechanisms) has generated important results. Reviewing existing mechanisms (in collaboration with Prof. Davide Martinenghi) \cite{27}, we have discovered, contrary to common belief, that many, though not all mechanisms are able to tolerate inconsistencies. They ensure that transactions preserve consistent data while not being disturbed by extant integrity violations. Thus, such mechanisms can be used also when integrity is unknown or compromised in favour of availability. We have developed formal, easily verifiable criteria for choosing tolerant mechanisms, and for avoiding those that are not inconsistency-tolerant. Optimisations and extensions of inconsistency-tolerant mechanisms have been discussed in other papers of ours \cite{24,25,26}. For task 8.2 (Development of integrity checking mechanisms for full replication strategies), we have revised the possible support for integrity checking in existing database replication protocols. We have shown \cite{48,49} that several protocols need to undergo some patching in order to correctly manage integrity constraints, and we have evaluated how such patches impact system performance. For task 8.3 (Integration in our middleware), we have devised

<table>
<thead>
<tr>
<th>Tier</th>
<th># Papers</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORE A’</td>
<td>3</td>
<td>[5, 8, 39]</td>
</tr>
<tr>
<td>CORE A</td>
<td>7</td>
<td>[13, 19, 24, 51, 43, 44, 50]</td>
</tr>
<tr>
<td>CORE B</td>
<td>19</td>
<td>[11, 12, 13, 16, 17, 11, 15, 18, 21, 22, 29, 27, 31, 32, 33, 38, 40, 51]</td>
</tr>
<tr>
<td>LNCS</td>
<td>9</td>
<td>[9, 12, 16, 28, 33, 37, 47, 49, 53]</td>
</tr>
<tr>
<td>Other</td>
<td>17</td>
<td>[10, 14, 17, 20, 23, 26, 29, 30, 33, 36, 42, 45, 46, 48, 52]</td>
</tr>
</tbody>
</table>

Table 2: Project publications
an approach that allows to reorganise the transaction completion order for reducing the probability of abortion due to constraint violations detected by integrity checking. To this end, we have proposed prioritised atomic multicast protocols \cite{40,41}, and have designed a middleware module that can rate transactions according to their likelihood of integrity violations. In \cite{41}, we have experimentally proved that transactions reordered according to their violation capacity rating reduces the amount of abort ions when the transaction arrival rates are high.

3 Result Indicators

As explained in the previous section, all project objectives have been achieved according to the suggested schedule presented in the project proposal.

In order to assess the relevance of the achieved results, we summarise in Table 2 how many papers (published either in conferences or journals, and included in the references section of this document) belong to each one of the categories proposed in the CORE rankings. Such ranking classifies the publications into 5 different tiers (A+, A, B, C, and L), being A+ the best one, and L the worst one. We have not been able to include all our publications in the references section due to space constraints; the remaining papers can be found elsewhere (ITI, UPNa).

Note that CORE rankings are not complete, and that several conferences and workshops have not been included in such rankings. Thus, nine of our published papers do not belong to any CORE tier and they have been published by Springer in its LNCS series. They have been listed as a LNCS Tier in Table 2. Recall that Springer LNCS was included in JCR in some years (e.g., 2004 and 2005), belonging to the third quartern of JCR journals using an impact-factor ordering, and that we have five other LNCS publications rated as CORE A and another one rated as CORE B.

All these publications also show that our two groups tightly cooperate in all project tasks, since most of such publications have been generated by researchers of both groups. So, we have presented them jointly in a single table. This also proves that our collaboration in this project has been fruitful and that no management trouble has been found. Each group is specialised in different concerns (specification and formal proofs in the University of Navarra, implementation and benchmarking in the ITI), and our joint work is able to enhance the overall results.

In the following subsections we summarise the results of each group in the remaining items suggested in the report template.

3.1 Research Training

**ITI Group.** In the context of this project, the training researchers of the ITI group have obtained a PhD thesis (Rubén de Juan Marín \cite{17}, in September 2008), five DEA\footnote{Diploma de Estudios Avanzados, a certificate that must be obtained once the student has passed all PhD courses, required to initiate his/her PhD thesis.} (María Idoia Ruiz Fuertes, Jerónimo Pla Civera, and Raúl Salinas Monteagudo, all of them in February 2008; Josep M. Bernabé Gisbert in February 2009; and Juan Carlos García Ortiz in March 2007), and the lecturers of our group have already supervised three Master Theses in the new
Master in Parallel and Distributed Computing that has replaced the doctoral courses of our previous Doctoral Programme.

Besides this, our group obtained a grant allowing the stay of some of our training researchers in relevant foreign research centres. Such grants have permitted the stay of José Enrique Armendáriz-Iñigo (at that time, contracted by ITI) in McGill University under the supervision of Prof. Bettina Kemme (from October 2006 to March 2007); Emili Miedes de Elías in the Universidade Nova de Lisboa from July 2007 to September 2007, under the supervision of Prof. Luís E. T. Rodrigues; Josep M. Bernabé Gisbert (from October 2007 to December 2007) and Rubén de Juan Marín (from January 2008 to May 2008), both in the Università della Svizzera Italiana, under the supervision of Prof. Fernando Pedone. We have collaborated with such groups, generating some publications [7, 13, 46] already discussed in previous sections. Moreover, María Idoia Ruiz Fuertes obtained the FPI grant associated to this project. Since such kind of grants permits also a research stay in foreign research centres, we have agreed with Prof. André Schiper her stay in the École Polytechnique Fédérale de Lausanne in the second semester of 2009.

**UPNa Group.** Respectively, in the context of this project one training researcher (José Ramón Juárez Rodríguez) has obtained a DEA and has done a research stay (from May 2008 to July 2008) under the supervision of Prof. Rui Oliveira in the Universidade do Minho where he was introduced to the GORDA replicated database system. On the other hand, José Enrique Armendáriz-Iñigo was working at ITI from June 2006 to June 2007 and did a short research stay at Universidade do Minho with Prof. Rui Oliveira (May 2008). Finally, Augusto Mauch Goya has pursued a Master Thesis in Computer Science, and part of it was published in [6].

### 3.2 Collaboration with other Groups

**ITI Group.** Besides the groups mentioned above regarding research stays, the ITI group has participated in the DeDiSys project of the 6FP of the European Union. In such project, it collaborated with part of the Distributed Systems Group of the Technical University of Vienna (Austria), led by Prof. Karl M. Göschka; and with the Real-Time Systems Laboratory of the Linköping University (Sweden), led by Prof. Simin Nadjm-Tehrani. We have already discussed part of the collaboration with such groups in previous sections, presenting our shared publications directly related to the CONDEP project aims.

Dr. Hendrik Decker, senior researcher of the ITI group, also maintains a tight collaboration with Prof. Davide Martinenghi, as already indicated above.

**UPNa Group.** We are currently collaborating with a couple of international research groups. We have started a joint research line in database recovery and dynamic distributed systems with Grupo de Sistemas Distribuídos at Universidade do Minho led by Prof. Rui Oliveira, some results have been submitted to several conferences and are still under review process. Some collaboration in the formalisation of correctness criteria of replicated database systems providing SI has been done with Prof. Bettina Kemme at McGill University. One of our joint works has been recently accepted as a journal paper [39], besides, we discussed how to face database recovery in a replicated system with SI replicas [7]. A study of consistency in transactional memories has been jointly carried out with the Institut de Recherche en Informatique et Systèmes Aléatoires group under the supervision of Prof. Michel Raynal that is published as a technical report [35]. Finally, a study of replication of business objects in SOA has been carried out with an external PhD student (Michael Ameling) of the Professur Rechnernetze group led by
Prof. Alexander Schill at the Technische Universität Dresden that has been accepted as a conference paper [1].

References


