

Adapting the FT-CORBA Replication Management Service for Large-scale Distributed Systems

Lau Cheuk Lung, Joni da Silva Fraga
Graduate Program in Applied Computer Science- PPGIA
Pontifical Catholic University of Paraná – PUCPR - Curitiba – Brazil
Departamento de Automação e Sistemas, Universidade Federal de Santa Catarina, Florianópolis, Brazil
e-mail: lau@ppgia.pucpr.br, fraga@das.ufsc.br

ABSTRACT

The FT-CORBA specification, published by Object Management Group [13], is not suited for supporting object replication in large-scale distributed systems. The inadequacy or the lack of definitions in this specification for scalability of fault-tolerant applications in those systems is the main motivation for this research. In this paper¹, we introduce the GroupPac, a free software that implements the FT-CORBA specification, and offers a set of extensions (including failure detection, group membership and group communication) that aims to face the scalability problem in large-scale distributed systems.

Keywords: CORBA, Fault tolerance, membership

1 Introduction

Nowadays, many distributed applications, which need high availability and fault tolerance, have considered the FT-CORBA [13] specification as the best alternative for adapting to the requisites of open systems [13, 5]. The Fault-Tolerant CORBA specifications define a set of object services and useful means for supporting the implementation of replication techniques in distributed object systems.

Doing a detailed analysis in the current FT-CORBA specification, we verify some conceptual difficulties when that specification is applied in large-scale distributed systems, such as Internet. The main difficulty is due to the asynchronous characteristics of the large-scale networks – asynchronous interactions over these networks are the main source for non-determinism. The FT-CORBA specification does still not present any objective abstraction for this class of systems [13]. The emphasized model in the specification is limited when the fault management service, replication management service, and group communication support execute in a large-scale context. Furthermore, and not less important, there is not discussing about how these solutions can be integrated or built into the FT-CORBA architecture, without that results a modification on any interface which already has been standardized by OMG.

In the literature are found many works (most of them were published before the FT-CORBA standardization) with

proposal for extending CORBA specifications seeking for a good solution to suit group communication services (or toolkits) in a CORBA middleware [5, 3, 12, 9]. Those experiences [5] (classified in three approaches: integration [11, 8], service [3, 6] and interception [12, 9]) try to assure the requisites for interoperability, portability, and performance. Because they are recent, there not exist in the literature a more detailed study about FT-CORBA specifications, and, much less, proposition or discussions about how adapts their concepts for large-scale environments.

This paper presents a proposal for extending the FT-CORBA fault management and replication management services for scalability. The focus of this paper is data/object replication management for high availability (and fault tolerance). The proposed model allows us to specify an asymmetric failure detection protocol and a group communication support, both suited for large-scale distributed systems. The group communication model proposed by GroupPac allows that each fault-tolerance domain may have a different group communication tool [1, 8, 14, 16].

This paper presents in the section 2 a summarized description of the FT-CORBA specifications. The GroupPac infrastructure is presented in the section 3. At last, in the section 4, we pointed out the main conclusions of this work.

2 The Fault-Tolerant CORBA Specification

The FT-CORBA [13] architecture consists a set of service objects that supply the basic functionalities for building fault-tolerant applications. According to these specifications, the *Replication Management Service* (RMS) is responsible for management group of objects (membership), allowing object replicas to dynamically join or leave (normal or fault) the group. RMS uses the *Object Group Manager* (OGM) for updating the membership list. RMS uses *Generic Factory* (GF) to create or remove replicas of a group.

In that procedure, the *Generic Factory* object interacts with local *Factories* for creating or removal of replicas in different machines of a distributed system. Furthermore, the *Properties Management Service* (PMS) maintains the fault-tolerance properties of each group of objects under RMS control. These properties define, basically, how each group must be managed and controlled by FT-CORBA services. For instance, one of the information maintained by PMS is the definition of which replication technique must be used by a group, which could

¹ This work is partially supported by CNPq (Brazilian National Research Council) through processes 481523/2004-9, 506639/2004-5 and 550114/05-0

be: *Stateless, Cold Passive, Warm Passive, Active and Active with Voting* [13]. In the *Fault Management Service (FMS)*, failure detection (or monitoring) is based on timeout mechanisms. In FMS is also defined the interfaces for failure notifications and failure analysis. Finally, in the *Logging and Recovery Management Service (LRMS)* are defined the mechanisms for state transferring and recovering of faulty replicas.

3 GroupPac Infrastructure

The *GroupPac* project [9] corresponds to a set of specific services for supporting fault-tolerant applications. GroupPac interfaces are FT-CORBA compliant. Furthermore, in relation to the standard, GroupPac provides a set of extensions and adaptations in order to support large-scale distributed systems. These extensions are added to CORBA in a transparent way for the applications, and without any modification on FT-CORBA interfaces.

3.1 Hierarchy of Fault-Tolerance Domains and Scalability

The literature indicates that the most appropriate way to treat the complexity of large-scale systems is the hierarchical decomposition of the problem [16]. In a large-scale context, with several groups composed by spread geographically objects, GroupPac introduces a management model and group communication model, both based on a hierarchy of fault-tolerance domains.

In GroupPac, it is proposed a model for structuring large-scale systems into a hierarchy of FT domains (see Figure 1). This hierarchy is constituted by two level of management: *Local FT-domains* and *Global FT-Domain*. The lower level is formed by Local FT-domains, where are managed the object groups. The Global FT-domain holds, exclusively, only the set of services responsible for assuring the interaction among the Local FT-domains. That Global Domain does not hold groups or subgroups of application objects.

Each domain holds its own fault-tolerance infrastructure (LRS, FDS, RMS, LNS, GNS e GSeq, see Figure 1) provided by GroupPac object services. The GroupPac services of a domain can only operate on groups and subgroups contained into this FT domain. The Failure Detection Service (FDS) is responsible for monitoring the hosts into its domain. When a faulty host is detected by FDS, it notifies to the *Replication Management Service (RMS)* of its domain, and so a new IOGR can be installed (with a new membership list). The IOGR of each group into a domain is, then, disposed by *Local Naming Service (LNS)* to all application objects of the system. A LNS contains all IOGRs (and also IORs) of the FT domain in which it belongs. The FDS, RMS and LNS groups can be replicated in any degree, and to be executed in any host of the domain. At last, the *Global Naming Service (GNS)* of the Global FT domain that makes the bindings among all Local LNSs of FT domains, allowing that objects of a FT domain can locate objects or object groups of other FT domains.

In this model, the domains may contain groups where all their replicas (or members) are controlled inside of their own domain. In GroupPac, these groups are called *Intra-domain Groups*. A group that has their objects spread in a large-scale network can be divided in several subgroups installed in different Local FT-domains. That is, instead of having a single domain for supporting a large group, we have then, a set of domains in order to support independently each subgroup of this large-scale group. Large-scale groups having member objects spread in several Local FT-domains are called *Inter-domain Groups*. In the figure 1, the groups A1 and A2 contained into the Local FT domain A are examples of *Intra-domain Groups*. The subgroups P1 (*Local FT domain A*) and P2 (*Local FT domain B*) form the *Inter-domain Group P*.

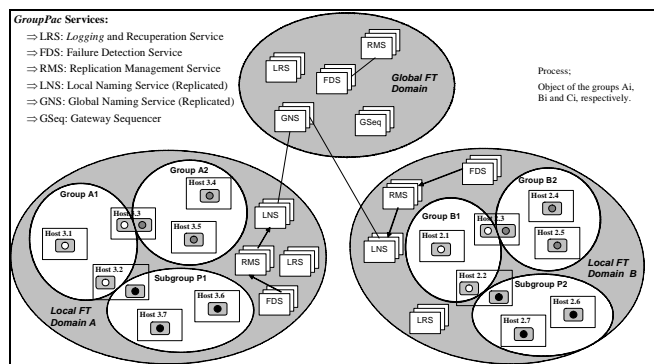


Figure 1. The hierarchical model for large-scale system.

In the figure 1, the FT Domain A contains and manages the group A1 and the subgroup P1. Thus, local domains may contain local groups (intra-domain groups), in which do not involve large physical distributions, and also subgroups of inter-domain groups. The global domain does not hold any application group or subgroup. Groups managed in this global domain are only formed by GroupPac object services.

Inter-domain groups, due their subgroups disposed in different local domains, make the management simpler and appropriated for scalability needs. This separation allows, for instance, that each domain may have own management and group communication protocols – established according to the characteristics of the runtime environment on which it is executing.

The membership changes are easier to be treated with this hierarchical decomposition in domains with subgroups. However, this decomposition is not a trivial solution because it implicates to define a set of supports and extensions on the FT-CORBA specifications that we will show in this section.

3.2 The GroupPac Failure Detection Service

The FT-CORBA failure detection, such as it was specified, is not suited for systems with asynchronous characteristics. In order to deal with these difficulties, we extended the notion of FT-CORBA failure detectors, but without alter the interface of that service. So, we assumed that a host is considered in crash if it does not respond to a certain number of failure detectors according to a specified timeout. Then, it is necessary an

agreement protocol to be executed by a set of detectors “to determine” a crash failure. The monitoring of a host from a set of detectors minimizes the probability of mistaken detection.

The solution that we adopted in this project was to specify a protocol based on majority vote to reach consensus about failure or not of an object. In our scheme, all detectors monitor all hosts inside a FT domain. The figure 2 exemplifies this detection scheme. The failure detectors adopt some ideas from the protocol proposed in [15, 14, 4]. For this case, ours detectors can be classified as *Perfect detectors* (class *P*, see [2, 4, 15]). Therefore, we assume that our detectors are always complete, after a timeout a faulty host (or process) is suspect by all detectors of a FT domain.

The failure detection service of the GroupPac is designed over two monitoring levels: *detectors level* and *hosts level*. On the first, the host failure detectors form a self-manageable group. That is, they control themselves for joining and leaving (normal or fault) of the *failure detectors* members of the group – each member monitors a partner (see figure 3). The monitoring of the failure detectors (FDi) is needed to indicate the number of members into the group for determining, for instance, the majority of detectors during a voting process.

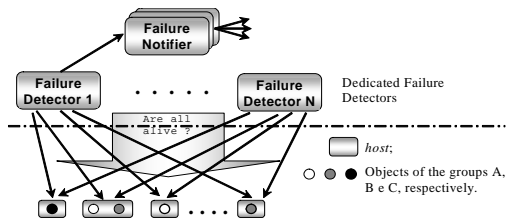


Figure 2. GroupPac failure monitoring.

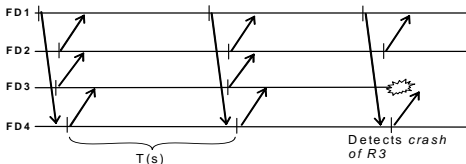


Figure 3. The failure detection service.

The monitoring at detectors level uses a centralized commit protocol (based on primary detector), in two phases (three phases in the worst case), to reach agreement among the members about the new composition of the failure detectors group (FD). This protocol is based on [15] and it was implemented in [9]. The members of this group compose a virtual ring and, periodically, each member monitors the partner immediately previous of the ring sequence. The coordination for obtaining of a new members list (membership) is centralized on primary failure detector FD1.

The second level of failure detection (*host level*) has the failure detectors group (FD1, FD2, FD3 and FD4 of the figure 3 and 5) monitoring the hosts (*Hi*) of the considered domain. When the crash of a host is detected, all processes (and objects) into this host are also considered as failed. All FDs periodically monitor, according to an interval *T(s)*, the same set of hosts of a FT domain (figure 2). All FDs decide, according to majority, if a particular host is faulty (crash) or

not (figure 4). If any detector suspects a host failure, the protocol executes a procedure based on majority vote, in order to reach consensus. The coordination of this consensus procedure is also centered on primary detector according to ring sequence.

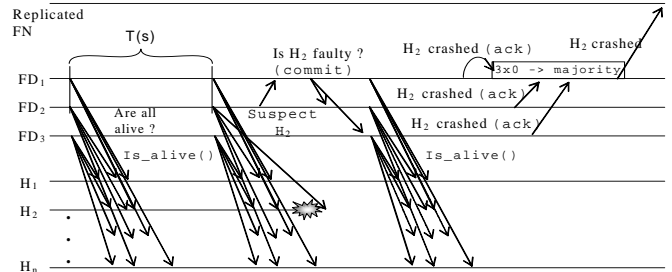


Figure 4. An instance of the failure detection protocol.

The figure 4 presents an instance of that protocol, FD2 suspects host H2 and it informs FD1 (the primary) about this suspect. Since then, it is initiated the protocol, in three steps, for the consensus and the agreement about failure of H2. In the first step, FD1 asks for all detectors (FD2 and FD3 of the figure 4), in the next monitoring interval (*T(s)*), that they report about the host H2 status (alive or failed). In the second step, after a new monitoring, each detector informs FD1 its opinion about the H2 status. At last, on the step three, the primary failure detector (FD1) decides, taking into account the received opinions, about the status of H2. After that, the primary FD1 informs, through the failures notifier (NF of the figure 4), to *Replication Manager* for it generates a new IOGR, removing H2 from list. The new IOGR is sent to the detectors group in order for they update the hosts list to be monitored.

3.3 Locating Object Groups on GroupPac

The proposed solutions in GroupPac are addressed in order for making easier the scalability. We looked for a form for allowing that objects, from different FT domains, were able to be located, and to allow them join or leave a group dynamically, during the lifecycle of the group. That is done through the association of FT domains with the naming service - it is important to emphasize that the FT-CORBA infrastructure does not provides an interface for making available IOGRs for client objects [13]. Therefore, the naming service is important for locating these IOGRs of intra-domain groups (groups A1, A2, B1, B2, C1 and C2 of the figure 1), and also of inter-domain groups (subgroups P1, P2 and P3 of the group P).

In GroupPac, the naming service also adopts the abstraction of domains hierarchy in order to compose a hierarchy formed by *Global Naming Service* (GNS group, in the figure 1) and by *Local Naming Services*. A LNS is responsible for managing all IOGRs (and IORs) of the application objects of a FT domain. The LNS also possesses, into its names context, a copy of GNS IOGR. The Global Naming Service (GNS) is responsible for managing the IOGR of each LNS of a domain – it possesses updated copies of LNS IOGRs of all registered

domains. The GNS group, which is part of Global FT Domain, allows the access to groups or subgroups of a FT domain starting from other FT domains (figure 1).

Furthermore, taking into account the naming service, as much LNSs as GNS both follow the CosNaming specifications [13] defined by OMG. In the conventional model, when a CORBA object needs access another object or a group, it obtains the IOR of naming service which contains the reference (IOR or IOGR) of the wanted group or replicated service. After that, with the reference, the wanted access is made. In GroupPac, for inter-domains interactions three levels of names resolution are needed in order to those IORs of application objects can be obtained.

3.3.1 Fault Tolerance for Naming Services of the GroupPac

The naming service, essential for binding objects, must assure fault tolerance requisites. Due to the disposition and different function of LNS and GNS into the system, we also have different FT properties attributed for these services.

Fault Tolerance for Local Naming Services

The *Local Naming Service* of each Local FT domain is implemented as a group of object services using the services provided by GroupPac (such as RMS, FDS and LRMS) in order to implement the primary/backup replication technique [9]. The LNS primary is responsible for binding and resolving all references of a FT domain. However, in order for a better performance, requests for resolving names (read operation) can be assisted by any LNS backups. For better accessibility, the LNS backups must be spread at different point (host) of a FT domain. When the failure (crash) of the LNS primary is detected (through the protocol presented in the item 3.2), a new primary is defined among the backups – the chosen is the first one defined by order in the IOGR – and homologated by RMS of that FT domain. With that, the new LNS primary must register its IOGR on the GNS, in order to become the new leader of the domain. More details about this implementation can be found in [9].

Fault Tolerance for Global Naming Service

For fault-tolerance, other solutions were adopted for the *Global Naming Service* (GNS). GNS has all IOGRs of the services contained into Global FT domain, and also all IOGRs of the LNSs of the Local FT-domains. Therefore, the GNS should be as more accessible as possible for all objects of each Local domain. In this case, the GNS group is implemented using active replication technique. We consider that in a global naming service, a request for obtaining IOGR of LNS is, basically, a read operation – it does not alter the GNS state. Furthermore, the requests to register new IOGRs, for different groups, can be handled in an independent way – a total order is not needed among these requests – what makes unnecessary to use a atomic multicast for this communications. Therefore, what we need is only to assure the order established by sender (FIFO ordering) when it does IOGR updates. A protocol for reliable multicast with FIFO ordering (FIFO multicast [7]) can

be used, which respond perfectly our needs. Operations for requesting an IOGR (resolve) do not need to be multicast to all GNS replicas because it is a read operation. One of the replicas, preferably the closest, can reply to that request.

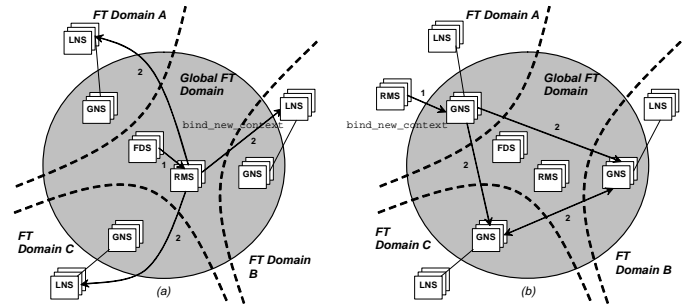


Figure 5. Naming service for supporting group management.

Due to those considerations, we present in the figure 5 the global naming service (GNS). Unlike the FT-CORBA specifications, where the objects cannot belong to more than a FT domain, on the other hand, the hosts do not present such restrictions (item 3.1). Despite the GNS replicas be part, exclusively, of the Global FT domain, they can be installed in hosts that pertain to the Local FT-domains, in order to make the application object access more efficient.

The GNS replicas form a group which is, just like LNS into local domains, managed by GroupPac services, which belong to the global FT domain. Due to large distance among GNS replicas, a longer timeout or failure detection protocols [2, 15] more appropriate for scalability should be considered. The FDS and RMS of this domain execute the task for keeping, through the IOGR, an updated member list (membership) of the GNS group.

Any change on the composition of the group members of GNS is generated a new IOGR, and sent by RMS to all LNS of each domain (the numbered arrows of figure 5a indicate the path of these information). In a similar way, when a new IOGR of LNS group is generated, the SGR of that local domain takes charge in sending a copy to a GNS replica, the closest of this local domain. This GNS replica registers and multicast the IOGR (using FIFO multicast) for the other GNS replicas of group. The numbered arrows in the figure 5b describe these actions involving the register of a new IOGR of the LNS group.

3.4 Group Communication on Hierarchical Model of the GroupPac

The hierarchical model for FT domains adopted by GroupPac allows each FT domain (Local or Global) has its own group communication support – the group communication tools can be different to each FT domain, but they must assure similar properties for agreement and ordering. The GroupPac allows three types of group communications: client communicating with an intra-domain group, both into the same FT domain; client communicating with an intra-domain group, each one into different FT domains; client communicating with inter-domain groups. The

first type, involving communications between client and groups belonging to a same domain is quite simple. It involves a client request to LNS to get the IOGR and so to connect to the target intra-domain group. The group communication of this type is implemented on GroupPac using interceptor mechanisms [13], and a proprietary group communication tool available such as object service on the considered domain.

When a client and an intra-domain group belong to different FT domains, the communication between both has to pass for a naming resolution a little bit more complex. The client object, in a domain A, when invoking an intra-domain group that pertains to a domain B, it must execute the following steps:

1. To access the local naming service (LNS) of its domain to obtain the reference (IOGR) of the global naming service (GNS);
2. To access GNS to obtain the LNS IOGR of the wanted FT domain, which contains the group that it wants to invoke;
3. At last, to access LNS of the FT domain B to obtain the reference (IOGR) of the target group.

The group communication in this case also uses interceptor and a proprietary group communication tool exposed such as object service. However, this object service and its correspondent group communication tool pertain to the FT domain of the group. A client to reach this group needs redirect its request through the Gateway Sequencer (GSeq), which retransmits the request to the group communication service of the FT domain where is the target group. We let the details of this mechanism for the next item. The third communication type mentioned above involves inter-domains groups and shall be discussed in the subsequent items.

3.4.1 Inter-domain Group Communication

The communication with inter-domains groups that has their replicas distributed in different FT domains, needs a support more suited than the types presented previously. In the literature are found several algorithmic solutions for group communication, including those with atomic multicast properties, where the groups are characterized by dispersion of their members in a large-scale network [16]. Usually, these researches solve the communication with these groups through the separation, into several subgroups, where the communication properties would be treated in a viable way, locally, into each subgroup. Furthermore, our proposal allows that each subgroup may have its own group communication tool, different from each other if they want to, but as long as the group communications properties (agreement and ordering) adopted by tool be the same defined for entire group (inter-domain group).

If we consider the FT-CORBA specifications and the propositions introduced by GroupPac, an inter-domain group would have, therefore, their subgroups distributed in different Local FT-domains. The figure 6 shows an example of an inter-domain group which will allow a better understanding about the dynamic of these groups. The objects group *P* of this figure is constituted by a set of subgroups P1, P2 and P3, distributed into the local domains A, B and C, respectively.

Each subgroup is managed and controlled independently by GroupPac services of each Local FT domain (item 4.1). Furthermore, when a new replica is inserted into one of the subgroups of *P*, by RMS of its FT domain, this new replica has its state updated by Logging and Recovery Service [13] of this domain. Therefore, the list of members (membership view) of an inter-domain group might be dynamic.

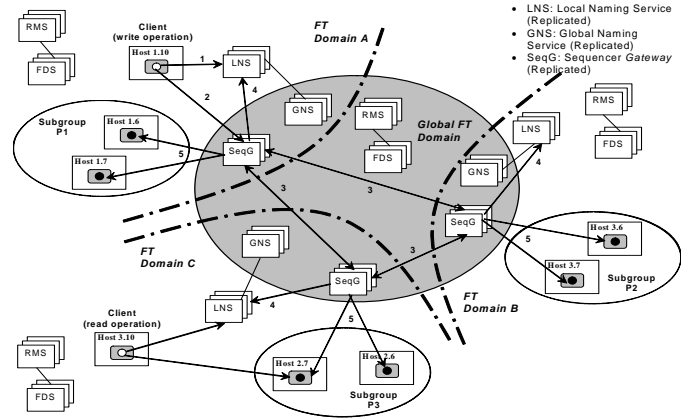


Figure 6. Group communication using GSeq.

In order to make possible the inter-domains group communication, it is proposed the following extensions in the FT-CORBA: (i) two new fault-tolerance properties; (ii) an extension in the IOGR structure; and (iii) a service that allows the communication with inter-domains groups according to the hierarchical model of domains.

3.4.1.1 FT Properties for Supporting Scalability

FT-CORBA properties (figure 7) are associated to each object group in a FT domain. These properties are managed by Property Management Service (Section 1). For example, among these properties are the replication type, fault monitoring style, initial number of replicas, fault monitoring interval, and interval for updating state. According to the specifications, these properties, and their values (parameters), can be extended according to the needs of the applications – the FT properties are additional information that, depending on the application, can be used or not during the execution of the system, and they are not an obligation in the FT-CORBA specifications. Therefore, to provide support for inter-domain groups, we proposed two new properties [10] (figure 7):

- ◆ *InterdomainGroup*: it determines whether a group contains members in different domains or not;
- ◆ *OrderingType*: it determines the ordering type to be used by Sequencer Gateway, Unreliable, Reliable, Causal or Total order. This property depends on type of ordering provided by group communication tool adopted by domain.

```

1. module CORBA {
2.   module FT {
3.     typedef sequence<Property> Properties;
4.     typedef sequence<FactoryInfo> FactoryInfos;
5.     typedef long ReplicationStyleValue;
6.     const ReplicationStyleValue STATELESS = 0;
7.     const ReplicationStyleValue COLD_PASSIVE = 1;

```

```

8.  const ReplicationStyleValue WARM_PASSIVE = 2;
9.  const ReplicationStyleValue ACTIVE = 3;
10. const ReplicationStyleValue ACTIVE_WITH_VOTING = 4;
11.
12. typedef long InterdomainGroupValue;
13. const InterdomainGroupValue NOT = 0;
14. const InterdomainGroupValue YES = 1;
15. typedef long OrderingTypeValue;
16. const OrderingTypeValue UNRELIABLE = 0;
17. const OrderingTypeValue RELIABLE = 1;
18. const OrderingTypeValue CAUSAL = 2;
19. const OrderingTypeValue TOTAL = 3;

```

} FT
Properties
Extension

Figure 7. Fault tolerance CORBA properties.

3.4.1.2 Extension on the IOGR structure

The CORBA specification defines that object or object group references, besides carrying information about the object location in the system, it allows add some additional information regarding the application type. Additional information is mapped through a data structure, called Tag. The tags can be added at the end of an object reference (IOR or IOGR) structure. Any information contained in a tag can, also depending on the application, to be read or not during the execution. In GroupPac, we extended the IOGR introducing a new Tag. This tag, called *TAG_FTProperties*, contains all FT-CORBA properties defined for the corresponding group – including the proposed properties in the previous item (4.4.1.1). It is important to emphasize that the inclusion of this tag is an extension to the IOGR. This extension allows to the client (through its message interceptor [13]), after obtain the IOGR, to know all FT properties of the group – including if it is or not an inter-domain group.

4 Conclusions

The GroupPac [10] were specified to allow the implementing of different algorithmic solutions to the failure detection service and group communication. For instance, for failure detection in the detectors group, an algorithm involving the unreliable detectors of [2, 4] could be used. On the other hand, the failure detection model to application objects has to be asymmetrical, due to the defined semantics for these detectors according to the FT-CORBA specifications. The hierarchical model of domains confines failure detection messages into FT domains, resulting in a smaller number of messages. In terms of related works, involving the use of the CORBA specifications for failure detection in distributed systems, the proposal presented in [3], although it does not use the concepts defined by FT-CORBA, it defines a set of detectors, called *WatchDog*, to monitor the objects of the application in a similar way to the detectors introduced by FT-CORBA. However, this work is limited to local network. In OGS [5] the failure detection is also similar to the specification, and their interfaces are different from FT-CORBA. However, OGS presents a consensus service that could be adapted for failure detection. Both proposals do not consider scalability aspects.

Basically, the set of extensions introduced here is due to the support for inter-domain group communication. The need to add two new FT properties (item 4.4.1.1) is due to IOGR [13] that limits a group into single FT domain.

Once the SeqG and GNS groups can be composed by few replicas, this allows us to abstract a little from a large-scale system model. The group communication support used in the domains involves service objects that encapsulate the functionalities of a group communication tool. In our development [10] the JavaGroup Toolkit was used. However, other tools, such as Isis [1] and Horus [14], could be used. The proposed hierarchical model also allows different group communication protocols to be used in the different domains.

REFERENCES

1. K. P. Birman,.: The Process Group Approach to Reliable Distributed Computing, Tr91-1216, Cornell Univ. Computer Science Department, Ithaca, N.Y., July 1991.
2. T. Chandra, S. Toueg,., Unreliable Failure Detectors for Reliable Distributed Systems, JACM, 43(1): 215-267, March 1996.
3. P. E. Chung, Y. Huang, S. Yajnik, D. Liang, J. Shih, DOORS: Providing Fault Tolerance for CORBA Applications, in poster of Middleware '98, Sept. 1998.
4. C. Delporte-Gallet, H. Fauconnier, and R. Guerraoui. "A realistic look at failure detectors". In Proceedings of the DSN'02, Washington - D.C. - USA, June 2002.
5. P. Felber and P. Narasimhan. Experiences, Strategies, and Challenges in Building Fault Tolerant CORBA Systems. IEEE Transactions on Computers, 53(5):497 - 511, 2004.
6. P. Felber and R. Guerraoui, "Programming with Object Groups in CORBA", IEEE Concurrency, Volume 8, Number 1, pp. 48-58, 2000.
7. V. Hadzilacos and S. Toueg,.: Fault-Tolerant Broadcast and Related Problems, In Distributed Systems, ACM Press (S. Mullender Ed.), New York, 1993, pp 97-145.
8. Isis Distributed Systems Inc, IONA Technologies, Ltd.: Orbix+Isis Programmer's Guide, Doc D070-00, 1995.
9. L. C. Lung, J. Fraga, J. Farines, M. Ogg, A. Ricciardi,.: CosNamingFT – A Fault-Tolerant CORBA Naming Service, Proc. of the 18th IEEE SRDS, Oct 1999.
10. L. C. Lung, J. S. Fraga, R.: Design and Implementation of the GroupPac, <http://groupac.sourceforge.net>, 1998
11. S. Maffei,.: Run-Time Support for Object-Oriented Distributed Programming, Ph.D. Thesis University of Zurich. Zurich, 1995.
12. L. E. Moser, P. M. P. Melliar-Smith, P. Narasimhan,.: Consistent Object Replication in the Eternal System, Theory and Practice of Object Systems, 4(2): 81-92, 1998.
13. Object Management Group, The Common Object Request Broker 3.0/IIOP Specification, OMG Document formal/2004-03-01, 2004. Available at www.omg.org,
14. R. V. Renesse and Kenneth P. Birman,.: Protocol Composition in Horus, Dept. of Computer Science of the Cornell University, Mar 1995.
15. A. Ricciardi and K. Birman, "Using Process Groups to Implement Failure Detection in Asynchronous Systems". In 10th ACM PODC. August, 1991.
16. L. Rodrigues, H. Fonseca and P. Verissimo, "Totally Ordered Multicast in Large-Scale Systems", In Proceedings of the 16th ICDCS, IEEE, 1996.