Design and Implementation of a HLA Inter-federation Bridge

Benoît Bréholée,
Pierre Siron
ONERA-CERT/DTIM
Information Modelling and Processing Department
2, av E. Belin, BP 4025
F-31055 Toulouse Cedex 4
Benoit.Breholee@cert.fr, Pierre.Siron@cert.fr

ABSTRACT: *In this paper, we discuss the design and implementation of a HLA inter-federation bridge. Our works are mainly motivated by the scalability and security problems, but we also consider the use of bridges for interoperability purposes. We describe several bridge topologies, including linear and cyclic inter-federations. We discuss problems raised by bridge federates and the use of different RTI implementations. We detail several solutions, leading to the design and implementation of a bridge prototype. Then we present our current results, and on-going works concerning performance improvements, interoperability, and security purposes.*

1 Inter-federations

1.1 Principle

Inter-federations are sets of interconnected federations. Events that occur in usual federations are not visible in other federations through HLA mechanisms since they are physically independant. Building an inter-federation consists in linking several federations so that events occurring in any federation may appear in the others. The existence of those associated events depends on the objectives and the design of the inter-federation. Events can appear as global to the inter-federation, but federations can have local or "private" parts which do not need to appear in other federations.

1.2 Interests

Our work on inter-federations is motivated by three main interests, related to the introduction of the notion of domain : mainly scalability and security but also interoperability.

1.2.1 Optimization and scalability

Inter-federations may be considered for optimization purposes. Although interconnection of federations has a cost, some configurations should take advantage of inter-federations. Links between federations can ensure a filtering role, to reduce the amount of data exchanges. In particular, middleware responsible for this link may define interest regions with *Data Distribution Management* services [1]. One interesting inter-federation structure is hierarchical federations [2], where some federates are actually subsytems made of federations. This architecture is particularly adapted to selectively hide information, which can reduce network traffic. Heterogeneous groups of federates with significant differences in event rates is another reason to consider the use of inter-federations [3].

1.2.2 Security

Having several federations for one simulation provides a practical topology to implement security aspects. The link between federations may not only transfer information but also take on data filtering. Trusted bridge federates can be realized to achieve this objective [4].

1.2.3 Reuse and interoperability

We are also interested in inter-federations for interoperability purposes, and particularly using different RTIs in one simulation. Once again with heterogeneous federations, one particular RTI implementation may be adapted to some groups of federates, while some other parts of the federation would be better used with another RTI. Inter-federations may optimize such situations. They are even mandatory in other configurations: for instance, when RTI and federates API conflict. Not all RTI implementations provide the HLA API in all its language bindings. Our RTI prototype CERTI [1] [5] currently only provides the C++ HLA API, and one application of inter-federations is to realize a simulation mainly based on CERTI and C++ federates, connected to a federation running a Java-compliant RTI.

Another aspect of interoperability focuses on the ease of reuse of federates. Some federations may be incompatible because of significant differences in FOM though the same concepts are used: joining the federates from both federations into one single federation would not work. An inter-federation link could be responsible for the translation of identical concepts that are expressed differently in several federations.

2 Federation bridges

Federations interconnection can be realized either through inter-RTI and/or inter-federate links. But connections between RTIs imply a completely new protocol, not provided by the HLA specifications. Actually, HLA principles focus on single-RTI federations, and the only convenient solution is to consider inter-federate links. Moreover since interoperabilty is essential, extensions to the current RTI services cannot comply with our objectives. That's why inter-federations are created at the HLA application level, with particular federates: *federation bridges*, or *bridge federates*.

The architecture of a simple inter-federation, based on one bridge, is described on figure 1: two federations are linked thanks to an inter-federation bridge. The role of this bridge is to observe each federation, and to reproduce its behaviour in the other one. Any object or interaction has to be represented by a surrogate

objet/interaction in the other federation. Each federation appears in the other one as a "proxy federate", managing "proxy objects", that represent original objects from the original federation.

Depending on the objectives of the inter-federation, bridges have to create proxy entities for all objects and interactions, or for a particular set. Bridges have to translate RTI-specific informations, that differ from one federation to another, but that represent the same entities: an attribute handle in one federation probably has a different value in its associated proxy federate's federation. The need for such a transformation manager has been described in [6].
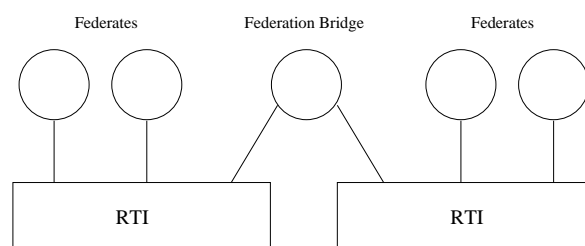

Figure 1: Simple inter-federation

3 Inter-federations topologies

Bridges simulate federations (or parts of them) into other federations. With many bridges (possibly having different behaviours), complex inter-federations can be implemented.

3.1 Linear inter-federations

The inter-federation designer is aware of the role of the bridge. But inside federations, federates do not have this information, since proxy federates appear like any other federate (federates shouldn't even have information about other federates, but only about shared entities such as objects and interactions). The same is true for the involved RTIs: a proxy federate is juste a federate involved in a federation. This is why every federation participating in an inter-federation is still a usual federation containing *all* the simulation objects, and where all the simulation events occur.[2]

---

[1] www.cert.fr/CERTI/

[2] except "private" objects/events voluntarily filtered

This is actually one goal of inter-federations: at the object/interactions level, the inter-federation should behave like the single federation based on the same federates.
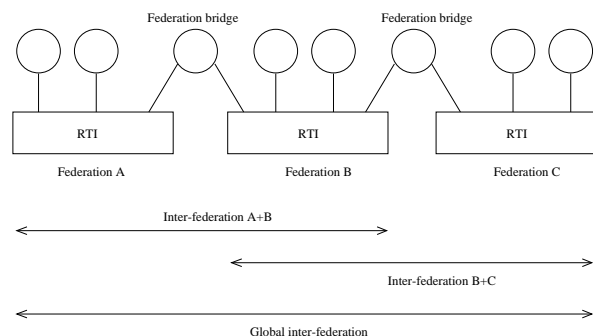

Figure 2: Inter-federation with 2-federation bridges

Then new bridges can be added to federations participating in an inter-federation: since any federation contains all the public simulation objects and interactions (either from its federates or from a proxy federate), a bridge connected to this federation can observe the whole inter-federation, and represent it through a proxy federate in a third federation. And then again, another bridge can be connected to any federation involved in this inter-federation (Fig. 2). In this case, any number of federations can participate to the inter-federation even if each bridge can only handle two federations.

Another way to extend our initial simple 2-federation inter-federation, is to have the bridge observe one federation, and simulate it in many other federations. On figure 3, only one bridge handles four federations. In this case, the bridge (and in particular its transformation manager) has to handle any number of federations.

These inter-federations are linear: there is only one way from one federation to another one. In this case, when one bridge resigns its federations, the (inter-)federations it used to connect become independant. It can be noted that in these linear inter-federations, bridges have to represent sets of objects and interactions, but there is no other constraint: any public object has to be represented by a proxy object in the connected federations.
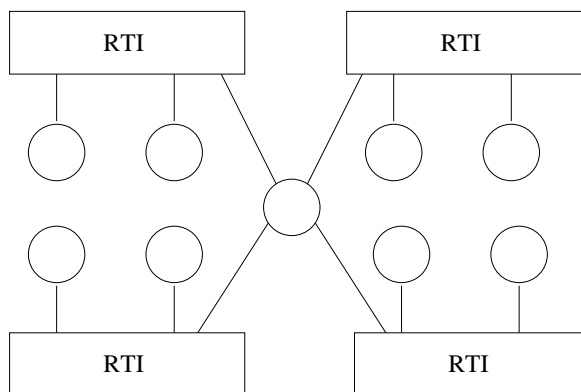

Figure 3: Inter-federation with an n-federation bridge

3.2 Cyclic inter-federations

Cyclic inter-federations have structures were several paths exist from one federation to another (Fig. 4). Obviously, bridges used for linear inter-federations do not work with this kind of configuration: if bridges systematically create proxy objects, one object will have several proxy objects in some federations as soon as there is more than one path from the source to the target federation. And actually, since redundant proxy objects are themselves represented through several paths, this leads to an infinite number of proxy objects [7].
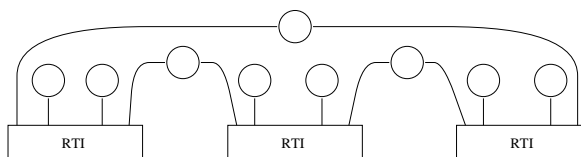

Figure 4: Cyclic inter-federation

This problem does not appear in some particular inter-federations. Federations can have private objects, when bridges do not create proxy objects for them. Actually the scope of such objects could be a few federations. For instance, if an object class only exists in two federations out of the three ones on figure 4, there is no possible cyclicity for objects of this class.

Our approach to inter-federations is based on this particular case: every federation entity should be

represented in the other federations through one path. Though the inter-federation has a cyclic structure, this cyclicity has to be broken somehere for each entity. This is really different from the linear inter-federations, since in this case different entities may be represented through different paths.

We already had considered filtering in bridges for security or optimization purposes. Now filtering is also useful to create cyclic inter-federations: bridges have to be configured so that in the inter-federation, entities are represented once and only once in each target federation.

3.3 Generalized inter-federations

All the previous kinds of federation bridges are compatible to each other, provided that bridges involved in cyclic paths are correctly configured. One big inter-federation could use the 2-federation linear topology in some part of itself, n-federation bridges in some other parts, and cyclic aspects elsewhere, as in the figure 5 example.
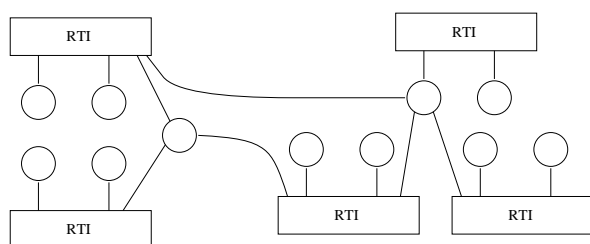

Figure 5: Generalized inter-federation

4 Problems and solutions

Inter-federations design and implementation raise several problems.

4.1 Information access

Inter-federations are created at the federate level, with federation bridges. These bridges do not have privileges to access RTI information: like every federate, bridges receive only *RTI Initiated* services, and can only send *Federate Initiated* services.

The HLA specifications provide a way for federates to receive information about other federates' actions. The MOM[3] provides objects and interactions reflecting the use of HLA services by other federates. With this MOM, it's possible for a federation bridge to guess the behaviour of RTI in some particular situations. One problem that RTIs may still have unexpected behaviours. Moreover, the fact that we are focusing on the use of CERTI, which currently doesn't provide MOM services, led us to consider the use of MOM as a solution to implement in the last resort.

Another solution is to add new HLA services, providing the necessary information. There are examples in [3] and [8]. The problematic aspect of this solution is that it represents an increase in the number of services a federate has to provide in a federation. Another problem is that, until such propositions are standardized and implemented in RTIs, interoperability cannot be ensured.

Finally, it's also possible to consider particular protocols between federates (e.g. with interactions) to broadcast information usually managed by the RTI. Then the constraint is a significant modification of existing federates; once again this solution doesn't comply with our objectives.

4.2 HLA mechanisms simulation

4.2.1 Unconditional mechanisms

Mechanisms such as *Register Object Instance/Discover Object Instance* †, *Update Attribute Values/Reflect Attribute Values* †, *Send Interaction/Receive Interaction* † are unconditional: unless the federate's request is not valid, the RTI has to send the reflect message corresponding to an update. Bridge federates subscribe to all attributes and interaction classes, receive the reflect messages, and can update the simulation in other federations: updating proxy attributes, sending proxy interactions, etc.

4.2.2 Global mechanisms

But there are global, conditional mechanisms: basically these are the services that require consensus from the federates, so that the RTI can provide a global status to a particular federation operation. The federation save

---

[3]Management Object Model

process is a typical case of global mechanism: federates are requested to carry out a save process, and to indicate whether the save process succeeded or failed. Federation restoration, federation synchronization, and some *Ownership Management* mechanisms are similar, though they differ in the details.

Such mechanisms may result in deadlocks if the bridge federates try to represent the global state of the federations they simulate. This is detailed in [6] and [8]. In the deadlock situation, the MOM solution consists in observing the federates, and considering a federation saved when all its "normal" federates succeeded in the save process. One risk is that problems in the RTI can still occur. Instead of using a MOM, we considered a bridge replying it's saved though the federation it represents is not saved yet. The deadlock disappears, and if all the federates succeed, the federation process succeeds in all the federations. Problems occur when one federation fails its save process: some federates may receive a *Federation Saved* † message, that is relevant only in their federation, and not at the inter-federation level. In this case, only the federates in the federation where the request was initiated can be ensured to receive a correct save process response: it is the only federation for which the bridge is going to make its response depend on the result of the other federations. So, one possible solution is to design an inter-federation in which such consensus processes are managed by only one federate: the federate sending the initial request, has to be the only one to use the result of the process.

4.3 Practical problems

4.3.1 Transformation manager

Though the inter-federation is only one simulation, each federation is independently managed by a RTI. Even if the FOMs are identical, RTIs may assign different handles to corresponding classes. This is bound to happen to dynamic entities such as objects. To correctly create proxy objects, bridges need to keep track of these associations between federations, and to translate federation-specific informations. This is the purpose of a Transformation Manager component [6].

4.3.2 Connection to several RTIs

HLA provides an API specification [9], but it doesn't contain specifications of compile-time details such as library names. This prevents the creation of generic bridges adapted to any HLA-compliant RTI. Then there is still an API-level problem in single process bridges, since both RTIs share the same API. Currently, we have access to the sources of our bridge prototype and of one RTI involved in our inter-federations (CERTI), so adaptation is possible, even in a single process bridge, with namespace changes. Concerning the lack of binary standard, this problem was raised and discussed during the 2002 Fall SIW RTI&C forum[4], and binary specifications should be proposed soon and eventually integrated in a future revision of the HLA specifications.

5 Design and implementation

5.1 Approach

Our approach to inter-federation design and implementation was directed by many practical aspects: the first set of HLA mechanisms to be translated through a bridge prototype was dictated by the involved RTIs, mainly CERTI. Since CERTI doesn't provide all HLA services, we focused on the currently available services, and didn't consider the use of MOM. This restriction was not a problem since the currently available services are the result of our current needs.

Then we considered the use of several methods previously evoked. Of course when possible, the bridge has to translate HLA mechanisms. In the problematic cases, we chose different solutions depnding on the concerned mechanism.

5.2 Inter-federation design

We did not want to restrict inter-federation topologies, to test several configurations involving a significant number of federations and bridges. Particularly, though we considered the principles of hierarchical federations, our interest relies more on cyclic inter-federations to decrease RTI load: in federations, performances decrease with the number of federates, and the main bottleneck is the RTI.

---

[4]this problem is also encounterd by vendors of proprietary federates

5.3 HLA mechanisms translation

Our bridge prototype uses *Federation Management* to join federations, but doesn't translate its other global mechanisms. The prototype uses *Declaration Management* to publish and subscribe to all entities found in respective FOM documents. Of course objects of any class missing in a target federation are not translated into this federation. Most *Object Management* services are implemented: unconditionnal mechanisms are the simplest translation case. Handles are translated through a transformation manager, but values and timestamps are not modified. Finally this prototype does not use *Ownership Management* and *Data Distribution Management* services: *Ownership Management* services are not essential to our needs. These services can be safely ignored in some federations: in this case ownership transfers only occur in one federation, and not at the inter-federation level. But ignoring these services would be a serious problem in inter-federations requiring ownership transfers between federates belonging to different federations. *Data Distribution Management* services cannot be directly translated by a federation bridge: all these services are *Federate Initiated*, and bridge federate cannot observe regions existing in one federation. The use of routing regions and *Data Distribution Management* services is still possible, but should be realized during the inter-federation design: regions can be used for optimization purposes, and particularly when ensuring cyclic inter-federations do not contain multiple paths for one particular entity.

6 Results

Our first result is the realization of inter-federations of different topologies. The bridge we have implemented connects any number of federations, and the use of several bridge instances gives access to all linear inter-federations. Not all the HLA mechanisms are translated, we mainly focused on *Object Management* services.

With the addition of filtering possibilities, other experiments were realized: creation of inter-federations with "private" objects, by the use of different FOMs. With filtering, design of cyclic inter-federations is possible, the only requirement being to forbid redundant proxy objects or interactions.

The first step concerned the realization of a bridge supporting various topologies. Then some performance tests were conducted with this bridge prototype. Table 1 summarizes those results. Each line represents one simulation based on 8 federates (not counting bridge federates), and indicates the number of time steps in a coordinated-time execution (duration: 1 minute). CERTI was used as RTI, and all the components were running on different hosts (federates, bridges, RTIG). All the federates were time constrained and time regulators, and were managing one object.

The single federation is not an inter-federation, and corresponds to one RTI and 8 federates. In the 2-federation tests, each federation had 4 federates, not counting the bridge, and in the second one all objects but one were private. The n-federation inter-federation refers to a unique bridge connecting 8 federations (each of them having only one federate). The cyclic inter-federation was realized with two federations of four federates, interconnected with two bridges, each of them representing two objects from each federation.

(inter-)federation type	time steps
single federation	*840*
two federations	562
two federations (with hidden objects)	587
n-federation inter-federation	1191
cyclic	369

Table 1: Time steps in various inter-federations

It should be noted that this values do not correspond to optimized inter-federations, they are the result of a bridge prototype that translates HLA mechanisms as defined in the previous sections. The bridge could unsubscribe classes not involved in the management of proxy objects: for example, the bridge subscribes to private classes, though target federations do not contain such classes.

In those inter-federation executions, the number of federates and objects was constant, and we used as many hosts as possible. So, while the nominal federation execution involves 9 hosts (8 federates and 1 RTI), the n-federation bridge inter-federation is based on 17 ones (8 federates, 8 RTIs, 1 bridge). The number of federates is a major factor in the decrease of performance, and the use of a bridge represents additionnal cost. It

appears that the use of one bridge, reducing the number of federates to 4 per RTI, is not enough to compensate the cost of the bridge. On the contrary, when the bridge connects several federations, letting only one federate per federation, global performances improve.

Since these experiments were realized with only one kind of object, most optimizations based on unsubscriptions would have been useless here: bridges and federates have to publish and subscribe to the object class. Bridges filtering data, and cyclic interfederations are designed to handle significant number of different objects and classes. This is probably why the results of their tests do not correspond to what was expected. Filtering did not affect significantly the 2-federation bridge. One explanation is that in this case, the number of publications and subscriptions is the same. Being subscribed to the unique object class, the bridge receives useless reflect messages. Finally in the case of cyclic inter-federations, the execution is slower. But one should remember that many filtering options exist for cyclic inter-federations, that are not used in this test. In this case, the benefit of two bridges consists in managing the translation of only two objects instead of four. The "private objects" test indicates the benefit is not significant here (the number of objects is low). This is probably why the addition of a bridge (which means the addition of a bridge *federate* in each federation) outweighs potential benefits.

These inter-federation executions indicate performance results for several kinds of topologies, but not all of them appear to be relevant. Linear inter-federation tests are the closest to practical use of the inter-federations they involve. In cyclic inter-federations, and linear ones involving private parts, the overall design and the optimization possibilities have to be taken into account.

7 Conclusion

We have described the principles of inter-federations, the particular aspects we are interested in, and the main approach to their implementation: federation bridges. We detailed several inter-federation bridge topologies, then we summarized several problems and common solutions. We presented the choices we adopted, based on practical constraints and on our interest in interfederations as enforcement of the notion of domain. The realized prototype initially aims at implement-

ing various inter-federation topologies, but optimizations specific to complex (and particularly cyclic) interfederations have to be implemented. Finally we described the results of some performance tests, indicating the effect of some topologies. Inter-federations involving private objects, and cyclic inter-federations could not take advantage of these kinds of executions though.

8 On-going works

Many on-going works concern the implementation of the bridge itsself. Not all the HLA mechanisms are translated through the bridge, and particularly the *Ownership Management* services have to be implemented. Some other mechanisms will probably not be considered until they are implemented in CERTI.

Another part of the bridge implementation concerns optimizations through the detection of irrelevant messages, and a better configuration of the bridge to support data filtering, particularly with *Data Distribution Management* services.

We are currently implementing inter-federations involving different RTIs (RTI-NG and CERTI), but a generic solution to this interoperability problem requires binary standards in the HLA specifications, and probably the implementation of a multi-process bridge.

Finally, one fundamental aspect of the coherence of the bridge is its transformation manager. It's currently only used to translate handles needed by *Object Management* services. But the names in the FOM could be translated too, so that federations using different names for the same concepts (e.g. coordinates) can be bridged without federate modifications. Translation could be applied to values, and time advancement too, to treat scale differences.

References

[1] Daniel J. Paterson, Erik S. Hougland, and Juan J. Sanmiguel. A Gateway/Middleware HLA implementation and the extra Services that can be provided to the Simulation.

[2] Wentong Cai, Stephen J. Turner, and Boon Ping Gan. Hierarchical Federations: An Architecture

for Information Hiding. In *Proceedings of the 15th Workshop on Parallel and Distributed Simulation, pp. 647-74*, 2001.

[3] T. Lake. Time Management Over Inter-Federation Bridges. In *Proceedings of the 1998 Fall Simulation Interoperability Workshop*, 1998.

[4] LouAnna Notargiacomo, Linda M. Schlipper, and William E. C. Ponn. The High-Level Architecture Multilevel Secure Guard Project. In *Proceedings of the 2001 Fall Simulation Interoperability Workshop*, 2001.

[5] B. Bréholée and P. Siron. CERTI: Evolutions of the ONERA RTI Prototype. In *Proceedings of the 2002 Fall Simulation Interoperability Workshop*, 2002.

[6] W. Braudaway. The High Level Architecture's Bridge Federate. In *Proceedings of the 1997 Fall Simulation Interoperability Workshop*, 1997.

[7] C. Damon, R. Melton, R. Allen, E. Bigelow, J. Ivers, and D. Garlan. Formalizing a specification for analysis: the HLA ownership properties. Technical report, CMU-CS-98-149, Department of Computer Science, Carnegie Mellon University, 1998.

[8] J. Dingel, D. Garlan, and C. Damon. Bridging the HLA: Problems and Solutions. In *Sixth IEEE International Workshop on Distributed Simulation and Real Time Applications*, 2002.

[9] DMSO. HLA interface specification version 1.3, 1998.

Author Biographies

BENOÎT BRÉHOLÉE was graduate from a french engineer school of aeronautics and computer science in 1999. Then he is a PhD student at ONERA. He is a member of the Design and Validation of Computer Systems research unit.

PIERRE SIRON was graduate from a french engineer school of computer science in 1980, and received his doctorate in 1984. Then he is a Research Engineer at ONERA and he works in parallel and distributed systems and computer security. He is a member of the Design and Validation of Computer Systems research unit.