Investigating the Consensus Problem in HLA Bridge

Jichuan Chang
Owen Cheng
Ganapathiraju Venkata Naga Satya Madhavi Kumari
Annie Luo

May 11, 2001

Introduction

The High-Level Architecture (HLA) provides a common architecture for distributed modeling and simulation. In its original form the HLA allows a number of simulations to be joined together into a federation using a single run-time infrastructure (RTI). Recently there has been an interest in joining multiple such federations together using a mediating unit, called the HLA Bridge.

Several problems exist with the HLA Bridge, as identified by Dingel, Garlan, and Damon's paper, *A Feasibility Study of the HLA Bridge*. These include:

- Selective addressing
- Consensus
- Federate failure
- Service barriers
- State/behavior assumptions
- Insufficient information

The paper also proposed some solutions for each of the problem classes listed above. However, neither the existence of the problems nor the applicability of the solutions has been formally shown to exist.

In this paper, we investigate the Consensus problem, demonstrate its existence, and elaborate on one of its proposed solutions by using a model-checking method. We use Wright, an Architecture Description Language (ADL), to specify the HLA Bridge architecture; translate the specification into CSP; and model-check it using FDR2.

The terminology and basic architecture of HLA Bridge are defined in Dingel et al.'s paper. Figure 1 illustrates the architecture of the basic system instance that we work with in this project. Although the RTI is shown as a box, it is conceptually a connector, joining the federate and surrogate components. The Bridge is likewise a connector, joining two federations, with internal structure containing a transformation manager (TM) and two surrogate components.

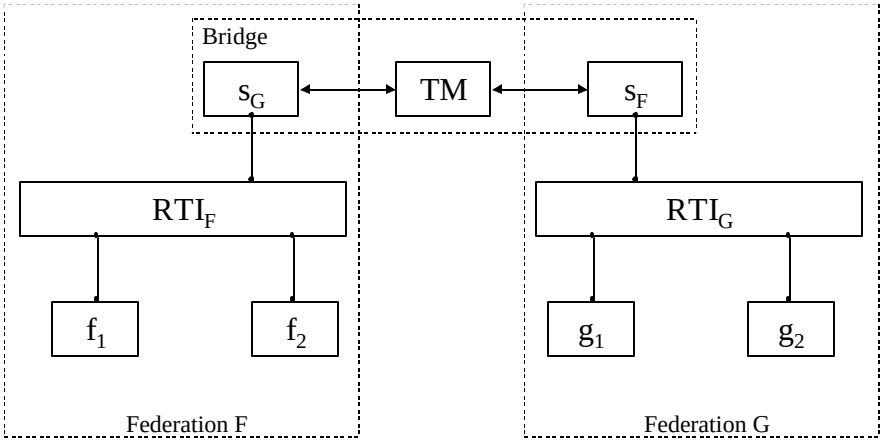

Figure 1. Federation F and G connected by a bridge
(*RTI represents the Run-Time Infrastructure,*
***TM represents the Transformation Manager in the bridge*)**

Problem Class Description

The importance of the Consensus problem lies in its widespread impact on the HLA Bridge. The feasibility paper states that the HLA uses four actions that require consensus: synchronization, save, restore, and time advance. The Consensus problem occurs after the invocation of a universal action request, *Everybody do A*, or a selective action request, *Some do A* (*set_of_Feds*), by a federate, when all the addressed federates have to achieve a common state by executing the action *A*. In this paper, we focus on showing the existence of the Consensus problem as applied to the *Federation Save* operation.

In the unbridged case, the RTI can invoke *Federation Save* directly on the addressed federates. Assuming that the federates notify the RTI when they've done *Federation Save*, the RTI knows when the designated federates have saved and the desired common state is reached.

In the bridged case, two problems arise. First, an RTI may not have direct access to all federates. Second, to be able to determine when all addressed federates have done *Federation Save*, an RTI requires a surrogate to know when all of the addressed federates that it represents have done *Federation Save*, and this kind of information is currently unavailable to federates. Any proposed solution must strive to keep surrogates as indistinguishable from modeling federates as possible.

Figure 2 depicts the process of the *Federation Save* operation. The circles with arrow lines indicate the flow of events as follows:

1. requestFedSave$_{(F)}$ – Normal federate f_1 initiates *Federation Save* operation
2. initiateFedSave$_{(F)}$ – RTI_F notifies all federates on Federation F, except f_1, to save
3. reqBridgeSave – Surrogate of Federation G (S_G) notifies G via the bridge to save

4. requestFedSave$_{(G)}$ – Surrogate of Federation F (S_F) receives *Federation Save* notification from across the bridge and then request Federation G to save, as if it is a normal federate
5. initiateFedSave$_{(G)}$ – RTI_G notifies all federates, except S_F, on Federation G to save
6. fedSaveComplete$_{(F,G)}$ – Normal federates notify their respective RTI of the completion of their save operation
7. bridgeSaved – ***Ideally***, one of the two surrogates tells the other that its corresponding federation has saved
8. fedSaveComplete$_{(SUR)}$ – The surrogates report to the RTI that they finished their save operation as if they are normal federates
9. federationSaved$_{(F,G)}$ – RTIs tell all federates that the federations have saved

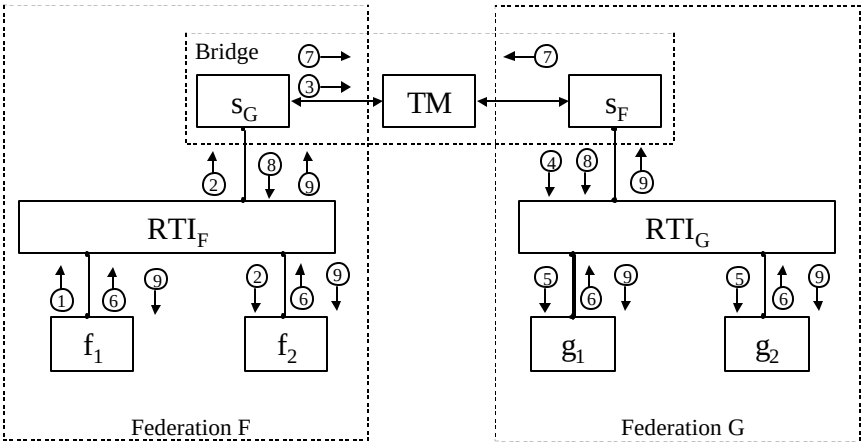

Figure 2. Event sequence for Federation Save operation

This sequence of events gets stuck on Step 7 and thus cannot complete successfully. Step 7 exposes the Consensus problem in the *Federation Save* process, in which both RTIs are waiting for the respective surrogates (e.g., RTI_F awaits S_G) to reply with the fedSaveComplete event. However, each surrogate is also waiting for its represented RTI (e.g., S_G awaits RTI_G) to notify it that the RTI has saved. Essentially, Step 9 needs to occur for either one of the RTIs before Step 7 can occur, but Step 9 cannot take place before Step 7 is performed on the other RTI.

This problem is shown to exist in FDR because of a deadlock in the refinement process. We now proceed to examine the solution proposed in Dingel, et al.'s paper.

Proposed Solution

The solution proposed by Dingel, et al. to resolve the Consensus problem for the *Federation Save* operation is to use a two-phased broadcast/collect scheme. This is achieved by adding two events to the set of events supported by the RTI and the federates, namely *letMeKnowWhenAllButMeHaveSaved* and *allButYouHaveSaved*.

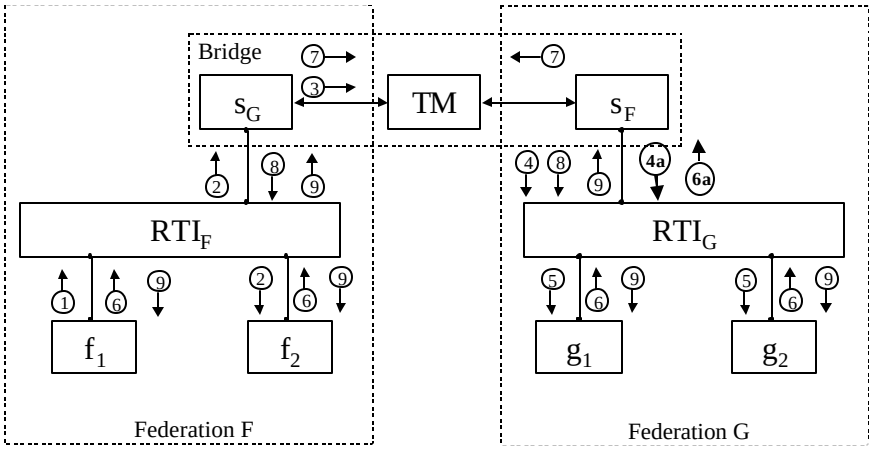

Figure 3. Event sequence for solution of Federation Save problem

In Figure 3 the two events are shown as 4a and 6a, respectively. After Step 4, in which the Surrogate S_F initiates the save operation on RTI_G, S_F tells RTI_G to notify it when all federates except itself in that federation have saved. After Step 6, RTI_G notifies S_F that all but it have saved. Thus, by Step 7, surrogate S_F knows that Federation G has saved, and can notify S_G on the other side of the bridge about this fact. In this way, the Consensus problem can be solved. Note that only the surrogate at the receiving end of the bridge participates in Steps 4a and 6a. One consequence of this restriction is that, if federates from both ends of the bridge initiate *Federation Save* at the same time, the bridge must delegate one of them as the receiving end of the request.

The other, less direct, consequence allows this solution to be extended to the situation in which 1) multiple federations are joined by binary bridges, 2) each federation is connected to at most two bridges, and 3) the federations are not connected in a circular fashion. This general case has not been verified by FDR checking, but the simpler case where two RTIs are connected by a bridge has been verified by FDR's fault-divergence refinement.

The original HLA specification was modeled in Wright by Allen, et al. [Ref: Allen]. This model was converted to CSP and extended to include bridge support. The original CSP model only had an instance of RTI with Federate roles, but no Federate component instances. In order to verify the bridge capability, new CSP processes emulating the behavior of the Bridge connector, the Surrogate component, and the Federate component have been added. An instance of the system reflected in Figure 4 was created for the purpose of FDR verification.

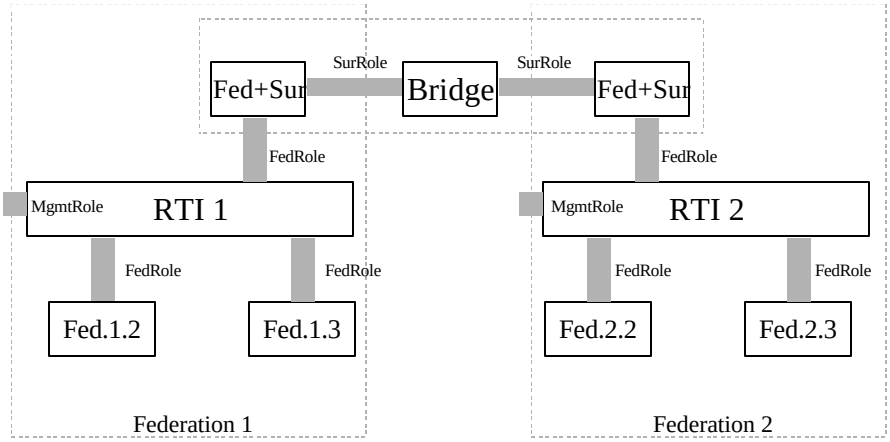

Figure 4. Diagram Representing the Instantiation of Wright model in CSP

Conclusion

This effort demonstrated the existence of one problem class in the set of problems identified by the HLA Bridge feasibility study, namely the Consensus problem. The proposed two-phased broadcast/collect solution has been shown to work through model checking in FDR, even though it is restricted by the assumptions laid out in the feasibility study. The other problem classes still need to be examined in more detail to determine whether HLA Bridge is really feasible. On the other hand, although the existence of one problem class and the applicability of the solution cannot demonstrate the overall feasibility of the HLA Bridge, it does give a ray of hope that the other problems can be overcome.

The original intention of this project was to extend the existing HLA with the new bridge architecture using the Wright ADL, then convert the Wright model to CSP specification, and verify it with FDR. Unfortunately, the wright2fdr converter core dumped on both our new model and the original HLA model. We were forced to manually translate the Wright model to CSP, a most tedious, albeit educative experience. The size of the CSP code was more than tripled in the process of instantiating the various components and the RTIs.

Although writing the CSP code was difficult, and FDR's debugging support was not particularly helpful, FDR was quite useful in checking the model, catching problematic process definitions, and, most important of all, showing that the proposed solution refines successfully.