

Capitolo 11

Appendice A (Breve introduzione ad HLA)

Introduzione

La comunità militare per anni ha sviluppato simulazioni distinte ed incapaci di interagire, la fine della guerra fredda e l'esplosione del fenomeno Internet hanno portato ad un ripensamento della situazione che si è concretizzato nella tecnologia High Level Architecture (HLA), nata basandosi sui concetti di interoperabilità e riusabilità considerati essenziali. Visto lo sforzo compiuto per la sua realizzazione si è fatto in modo di non creare un framework utilizzabile solamente per gli scopi di simulazione militare bensì una struttura che possa essere di supporto a tutta la simulazione distribuita. Questa tecnologia recentemente ha raggiunto il livello di standard IEEE (1516, 1516.1, 1516.2).

Struttura

HLA riveste il ruolo di collante al fine di ottenere una simulazione composta partendo da un insieme di simulazioni disgiunte. Questa macro-simulazione viene definita *federazione*, le singole simulazioni *federati*. L'entità minima di rappresentazione è costituita dal federato e non dalla singola entità simulata. La tecnologia HLA rappresenta un'architettura per simulazione basata sui componenti, in questa visione i componenti sono rappresentati dai singoli federati, ovvero simulazioni che hanno la possibilità di interagire tra di loro.

Nella pratica una federazione è composta da questi elementi essenziali:

1. un software che funga da supporto: il Runtime Infrastructure (RTI);
2. un modello per lo scambio di dati tra federati all'interno di una singola federazione: il Federation Object Model (FOM);
3. un certo numero di federati.

Questa struttura a componenti permette una facile decomposizione e ricomposizione di simulazioni oltre alla possibilità di combinare queste simulazioni con entità che sono ancora inesistenti al momento della loro creazione. Vengono

gestite strutture che non richiedono alcun cambiamento in presenza di modifiche di simulazioni o del loro aggiornamento e aggiunta di nuove funzionalità, in pieno accordo con i principi di riusabilità. Allo stesso modo l'isolamento tra simulazione e la presenza di interfacce standardizzate permettono di proteggere lo "strato simulativo" dall'evoluzione tecnologica dei mezzi usati per la computazione e della struttura di rete.

Comunicazione

La comunicazione tra federati può avvenire attraverso due sistemi distinti:

- oggetti;
- interazioni.

I primi hanno lo scopo di rappresentare entità durature nel tempo, sono organizzati in una struttura ad albero e la loro "comunicazione" avviene essenzialmente attraverso il classico sistema di pubblicazione/sottoscrizione. Le interazioni invece sono utili per rappresentare singole modificazioni dell'ambiente simulativo, non destinate a ripetersi invariate nel futuro.

Time management

Il concetto di time-management risiede tutto nell'assicurare che gli eventi generati tra federati siano consegnati nell'ordine corretto, indipendentemente dalla natura distribuita della federazione e dei conseguenti ritardi di rete. HLA prevede ben 4 tipi diversi di gestione del time-management:

- assenza di qualsiasi coordinamento;
- sincronizzazione conservativa (*pessimistica*);
- sincronizzazione *ottimistica*;
- activity scan (*event-driven*).

All'interno di queste possibilità ogni singolo federato può essere definito come time-constrained e/o time-regulating, ottenendo 4 ulteriori combinazioni. Ovviamente non tutti i federati sono obbligati alle medesime scelte, un federato che implementi solamente un meccanismo di log e verifica dei risultati non ha alcun motivo per essere time-regulating. La forma base di simulazione risulta essere quella basata su assunzioni pessimistiche, nel caso ottimistico l'infrastruttura HLA mette a disposizione le API necessarie per la sua gestione e per l'annullamento di eventi ma rimane compito dell'implementatore di ogni singolo federato la gestione del rollback di stato. Il lookahead riveste un ruolo fondamentale nella gestione del time-management, ogni federato definisce un proprio valore che entro certi vincoli può essere modificato anche in fase di runtime. La presenza di un lookahead pari a 0 porta alla simulazione event-driven indicata nella lista precedente.

Runtime Infrastructure (RTI)

L'RTI è la componente software che si occupa di permettere l'interazione tra i vari federati, tra i suoi obiettivi vi sono quelli di implementare l'insieme di regole definito dallo standard HLA e offrire una serie di servizi ai federati. L'implementazione risulta piuttosto "libera", soprattutto nella gestione della tecnologia di rete e questo può facilmente portare a prestazioni diverse a seconda dell'implementazione e delle diverse configurazioni di utilizzo.

Fino a questo momento l'implementazione di riferimento è stata quella finanziata direttamente dall'ufficio per la simulazione (DMSO) del Ministero della Difesa USA. La speranza che anche il mondo commerciale si occupasse di questo aspetto ha portato ad un'implementazione europea da parte della Pitch (una società svedese). L'implementazione DMSO appare sicuramente più matura ma anche piuttosto esigente in termini di risorse, quella Pitch (pRTI) interamente scritta in Java offre il vantaggio della portabilità. Il successo della prima è stato decretato soprattutto dalla possibilità di ottenerla liberamente dopo registrazione, la seconda è invece puramente commerciale. Si ha notizia di qualche altro tentativo di implementazione ma il loro stato dovrebbe ancora essere piuttosto embrionale.

Possibili ambiti di sviluppo

Oltre al settore principe di utilizzo per la simulazione militare vi sono una serie di altri campi d'interesse:

- interazione con tecnologie di middleware (ad esempio Jini e Corba) per lo studio di tecniche adattive mirate all'ottimizzazione dello speedup di simulazioni distribuite, per la riduzione delle necessità di management delle simulazioni, per la realizzazione di simulazioni dinamiche ed adattive rispetto alle risorse localmente disponibili;
- simulazioni ludiche (giochi in rete) e finanziarie. Questo porta anche alla necessità di un'implementazione dell'HLA runtime che abbia richieste limitate sia in termini di memoria che capacità elaborativa;
- integrazione con sistemi real-time;
- studio di scalabilità della tecnologia in presenza di un numero estremamente alto di federati;
- creazione di RTI con centralizzazione ridotta al minimo, ridondanza, gestione dei problemi di fault-tolerance.