



F.O.A.M.

Free Object Access Method

Un'introduzione

Il protocollo FOAM.

FOAM (Free Object Access Method) è un protocollo (un insieme di regole) che descrive come trasferire informazione tra diversi applicativi ed invocare procedure o metodi remoti.

Dalla definizione appena esposta, si evince che FOAM risponde a due necessità:

1. trasferire informazione;
2. dare la possibilità ad un applicativo di invocare procedure e/o metodi propri di un altro applicativo, ad esso eventualmente remoto.

Lo scopo di questo documento è analizzare i due punti precedenti.

Un'introduzione al concetto di trasferimento dell'informazione.

All'interno di un elaboratore elettronico, l'informazione è rappresentata attraverso circuiti "bistabili". Un circuito può trovarsi in uno solo di due stati possibili. In genere si associa il numero "0" ad uno stato, ed il numero "1" all'altro stato. Quando un circuito si trova nello stato "0" non consente il passaggio di corrente; viceversa, quando si trova nello stato "1", consente il passaggio di corrente. Ovviamente esiste anche la logica inversa di funzionamento: lo stato "0" consente il passaggio di corrente, mentre lo stato "1" non ne consente il passaggio. E' possibile far "commutare" un circuito, in modo da fargli cambiare stato. L'informazione più elementare è quindi rappresentata da un unico circuito, ed il suo stato interno può valere "0" oppure "1". Un'informazione così semplice si chiama "bit", dall'acronimo **binary digit** (cifra binaria). Informazioni complesse sono rappresentate attraverso una sequenza di bit, che corrisponde ad una sequenza di circuiti elettronici bistabili.

Ebbene, l'informazione gode di una serie di proprietà, tra le quali una in particolare in questi anni sta assumendo un ruolo molto importante: la trasferibilità.

Nei primi calcolatori elettronici, e fino alla fine degli anni '70 del secolo scorso, l'informazione veniva trasferita attraverso le "schede perforate": dei cartoncini della grandezza approssimativa di 1 dollaro americano, sui quali venivano prodotti dei fori, posti in corrispondenza biunivoca con i circuiti elettronici in cui era conservata l'informazione. Le schede perforate prodotte da un elaboratore elettronico venivano immesse in un altro elaboratore, e l'informazione in esse contenute passava dal primo al secondo elaboratore.

Oggi la scienza delle comunicazioni elettriche ha fatto passi da gigante e sono possibili metodi alternativi (e più evoluti...) per il trasferimento delle informazioni, caratterizzati da maggiore efficienza, velocità ed un più elevato controllo degli errori.

A questo punto è importante definire il concetto di "canale" di comunicazione: si intende per canale il supporto fisico attraverso cui l'informazione è trasferita. Nell'esempio precedente, il canale era costituito dal pacco di schede perforate; mentre ai giorni nostri i canali maggiormente utilizzati sono il doppino telefonico (per le comunicazioni tramite modem), il cavo coassiale (reti locali di modesta entità) e quello a fibra ottica, le onde elettromagnetiche, ecc.

Il trasferimento dell'informazione in FOAM.

FOAM consente di trasferire informazione in "multicanalità". Questo vuol dire che non si limita ad un solo canale di comunicazione, ma ne supporta diversi.

Con FOAM è possibile trasferire informazione attraverso la rete Internet (e-mail, ftp, web, ecc.), mediante la rete di telefonia mobile GSM (messaggi SMS, tecnologia Wap, ecc.) e tramite fax. E' anche possibile combinare tutti questi canali. Ad esempio, un'informazione giunta dalla rete Internet, può essere recapitata tramite un messaggio SMS.

FOAM mette a disposizione degli sviluppatori una serie di metodi che implementano il trasferimento dell'informazione. Tra questi:

- FOAM::SendFile() per spedire un file ad un elaboratore remoto;
- FOAM::GetFile() per comandare una richiesta di file da un elaboratore remoto;
- FOAM::SendSMS() per spedire un'informazione testuale attraverso un SMS, ad un telefonino GSM o ad un elaboratore munito di modem-GSM;
- FOAM::SendFax() per spedire un'informazione testuale ad un apparecchio fax (reale o virtuale) remoto o ad un elaboratore munito di modem-fax.

Un'altra caratteristica, molto gradita agli sviluppatori, consiste nella possibilità di effettuare l'aggiornamento del software da remoto. In pratica, quando un team di sviluppo software produce una nuova versione di un programma (ad esempio per eliminare malfunzionamenti, o per introdurre nuove caratteristiche), ha la possibilità di effettuare in maniera automatica l'upgrade della base di prodotto installato, senza doversi recare fisicamente presso ogni singolo cliente.

E' interessante notare che FOAM supporta una caratteristica nota con il nome di "fault tolerance", a livello di protocollo. Si dicono "Fault-Tolerant" (o tolleranti ai guasti) i sistemi capaci di non subire fallimenti anche quando avvengono guasti hardware o software.

Invocazione di procedure e/o metodi remoti.

L'altra caratteristica di FOAM, oltre al trasferimento dell'informazione, è quella di fornire la possibilità ad un applicativo di richiamare funzionalità proprie di un altro applicativo, eventualmente ad esso remoto. Questo apre lo scenario alla cosiddetta "elaborazione distribuita". E' possibile disegnare l'applicazione complessiva in maniera tale che il processo di invocazione remota sia completamente trasparente per lo sviluppatore. Non solo, un'applicazione può avvalersi di entrambe le caratteristiche: trasferire informazione in multicanalità ed invocare procedure o metodi remoti. La richiesta di invocazione di una procedura o un metodo remoto può avvenire anch'essa in multicanalità.

La possibilità di invocare procedure e/o metodi remoti, colloca FOAM nella categoria dei prodotti "middleware". Tramite FOAM è possibile disegnare applicazioni a 2 livelli (corrispondenti al classico modello client/server), ma anche a 3 o più livelli, consentendo alla logica applicativa di essere completamente svincolata dallo strato di presentazione (l'interfaccia utente). Esistono diversi sistemi e tecnologie "middleware", ma la peculiarità di FOAM è che esso supporta nativamente la multicanalità.

FOAM consente di utilizzare una sintassi Object Oriented per meglio integrare i metodi remoti all'interno della gerarchia di classi definite dall'utente, tramite i moderni linguaggi di programmazione, che supportano tale paradigma.

Esistono tecnologie correlate, come SQL Embedded, attraverso cui è possibile accedere in maniera trasparente ad un database remoto. Ad esempio è possibile costruire un'architettura a 3 livelli in cui al livello 1 c'è un RDBMS, al livello intermedio un Application Server (ad esempio Anaconda) ed al terzo livello un applicativo client che si occupa della logica di presentazione. Non è questa la sede per motivare la preferenza verso architetture multilivello rispetto a quelle a soli 2 livelli (client/server). Ebbene, è possibile interfacciarsi con il RDBMS astraendosi completamente da esso, tramite una semplice chiamata del tipo:

```
FOAM::SQL( "select * from table_1 where id > 5" );
```

Questa istruzione genera una chiamata remota verso l'Application Server (che è effettivamente interfacciato con il RDBMS), che invia a sua volta la stringa SQL, eventualmente pre-processandola. Il prodotto RDBMS restituisce i dati all'Application Server, il quale li invia al client che aveva originato la chiamata SQL.

FOAM si presta molto bene all'implementazione di Application Server (ed Anaconda ne è un esempio), infatti, oltre al già citato supporto fault-tolerant, prevede, sempre a livello di protocollo, il "load-balancing" per cui, in presenza di più istanze di Application Server, il carico viene ripartito in maniera ottimale, al fine di migliorare l'efficienza complessiva del sistema.