

Simulazione Parallela e Distribuita

Data Distribution Management (DDM)

Prima parte



Gabriele D'Angelo

gda@cs.unibo.it

<http://www.cs.unibo.it/~gdangelo>

Dipartimento di Scienze dell'Informazione
Università degli Studi di Bologna

Sommario

- Assenza di stato condiviso
- Cenni storici
- Problematiche principali
- Concetti fondamentali e terminologia
 - Namespace
 - Description Expressions
 - Interest Expressions
- Esempio: HLA Declaration Management
 - Class-based filtering

Propagare le informazioni

Un simulatore genera nuove informazioni che possono essere d'interesse per altri simulatori, ad esempio informazioni sullo stato di entità:

- come vengono propagate queste informazioni?
- quali sono i destinatari dei messaggi di aggiornamento?
- SIMNET, DIS (in alcune versioni)
 - Broadcast
 - Filtering sulla ricezione dei messaggi
- Problemi?
 - Scalabilità
 - CPU
 - Overhead di comunicazione

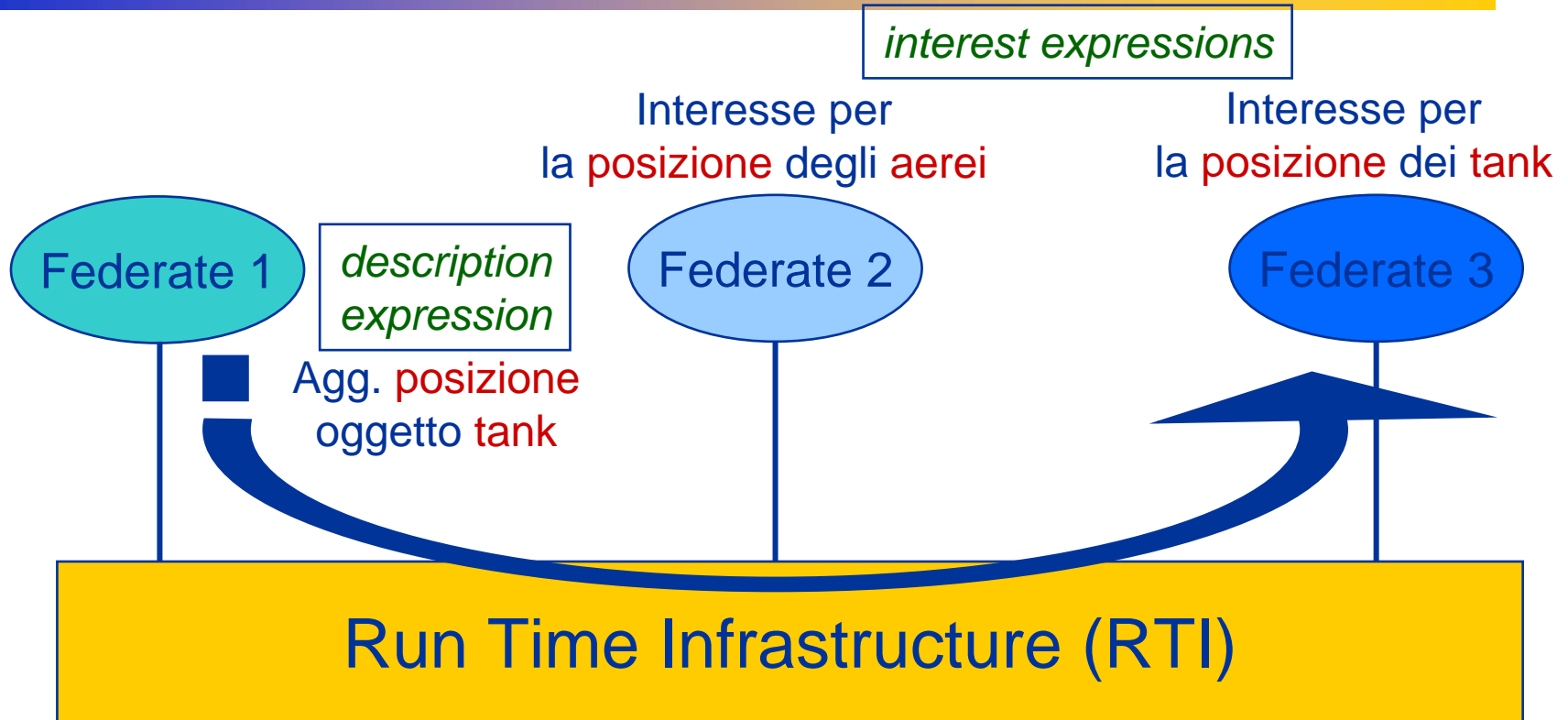
Data Distribution

Obiettivo: propagare le informazioni generate da un simulatore **solo** ai simulatori che sono interessati ai dati, cercare di evitare le spedizioni inutili

Cosa implica?

- Un modo per i simulatori di specificare a quali informazioni sono interessati per la ricezione (**interest expressions**)
- Un modo per descrivere le informazioni prodotte (**description expressions**)
- Un linguaggio comune (vocabolario) per definire interest & description expression (**name space**)

Esempio



Analogia: i newsgroup

- Description expression
 - Nome del newsgroup dove il messaggio viene “postato”
- Interest expression
 - Insieme dei newsgroup ai quali si è “sottoscritti”
- Name space
 - I nomi dei newsgroup

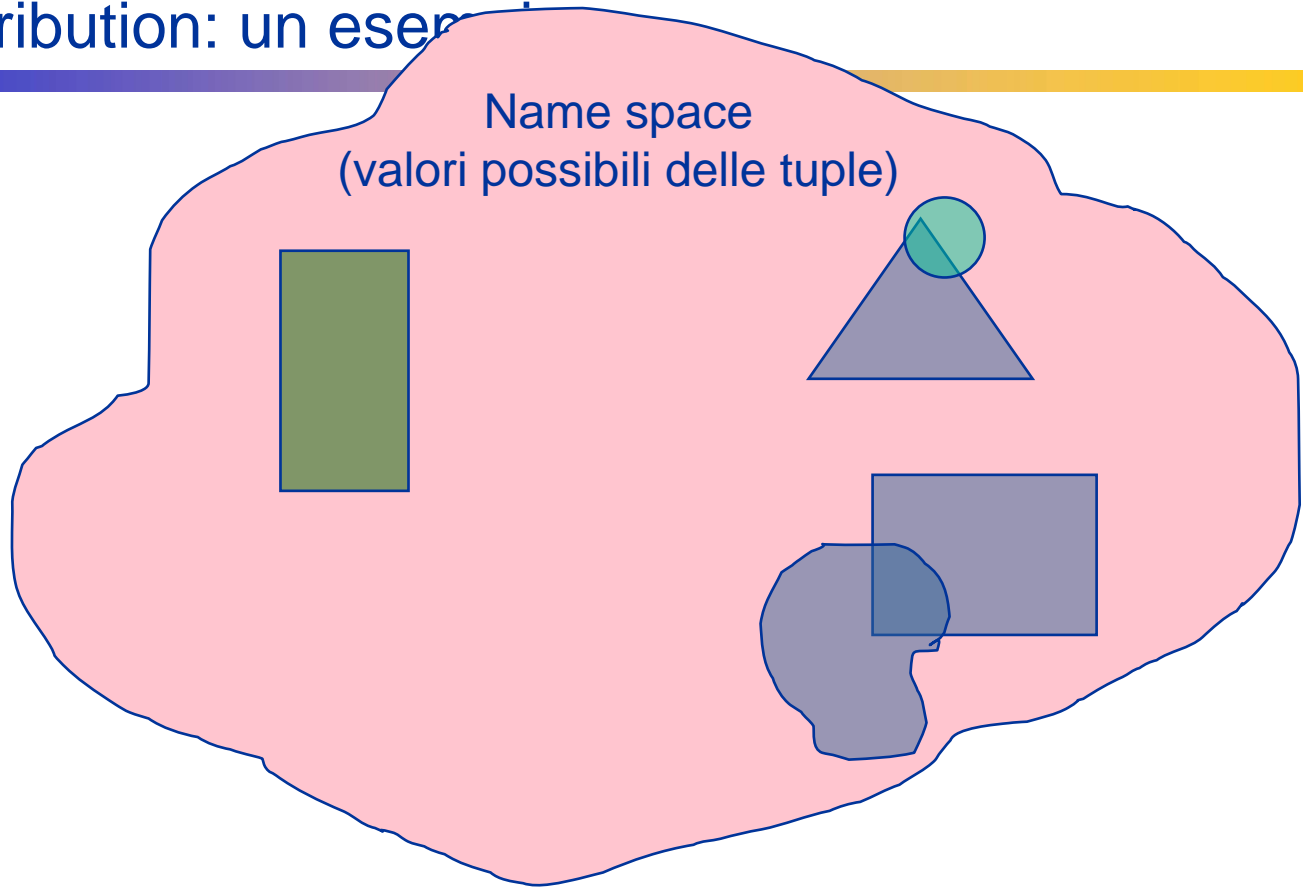
Name Space




- “Vocabolario” necessario per definire:
 - Data Description Expressions
 - Interest Expressions
- Insieme di tuple (V_1, V_2, \dots, V_N) dove V_i è di tipo base o tupla
 - Esempio: (class, location)
 - *Class: di tipo <tank, aircraft, ship>*
 - *Locazione: tupla (int: X-coordinata, int: Y-coordinata),
dove $0 < X < 1000$ e $0 < Y < 1000$*
 - Esempio pratico: (tank, (30,200)); (aircraft, (10,20))
 - I valori delle tuple possono riflettere proprietà statiche degli oggetti oppure dinamiche (valore di attributi)

Interest & Description Expression

- Interest Expression: sottoinsieme del namespace
 - Interesse per tutti gli aerei
 - $(aereo, (X, Y))$ per ogni X e ogni Y
 - Interesse per i tank "vicini"
 - $(tank, (X, Y))$ con $10 < X < 20$, e $130 < Y < 150$
- Description Expression
 - $(tank, (15, 135))$
 - $(aereo, (X, Y))$ con $35 < X < 38$ e $98 < Y < 100$
- Data routing
 - Un simulatore riceve l'informazione se la "description expression" (del dato) è sovrapposta alla sua "interest expression"

Data Distribution: un esempio



-  Interest Expressions, Simulatore 1
-  Interest Expressions, Simulatore 2
-  Description Expression di un messaggio

Il messaggio è propagato al Simulatore 2, ma NON al Simulatore 1.

Approccio statico o dinamico

■ Static Data Distribution

- Le proprietà presenti nel **namespace** sono statiche e non cambiano durante l'esecuzione della simulazione
- Esempio: il Declaration Management in HLA
 - *Data Distribution basata su Classi*
 - *Sottoscrizione all'attributo posizione di tutti gli oggetti "tank"*

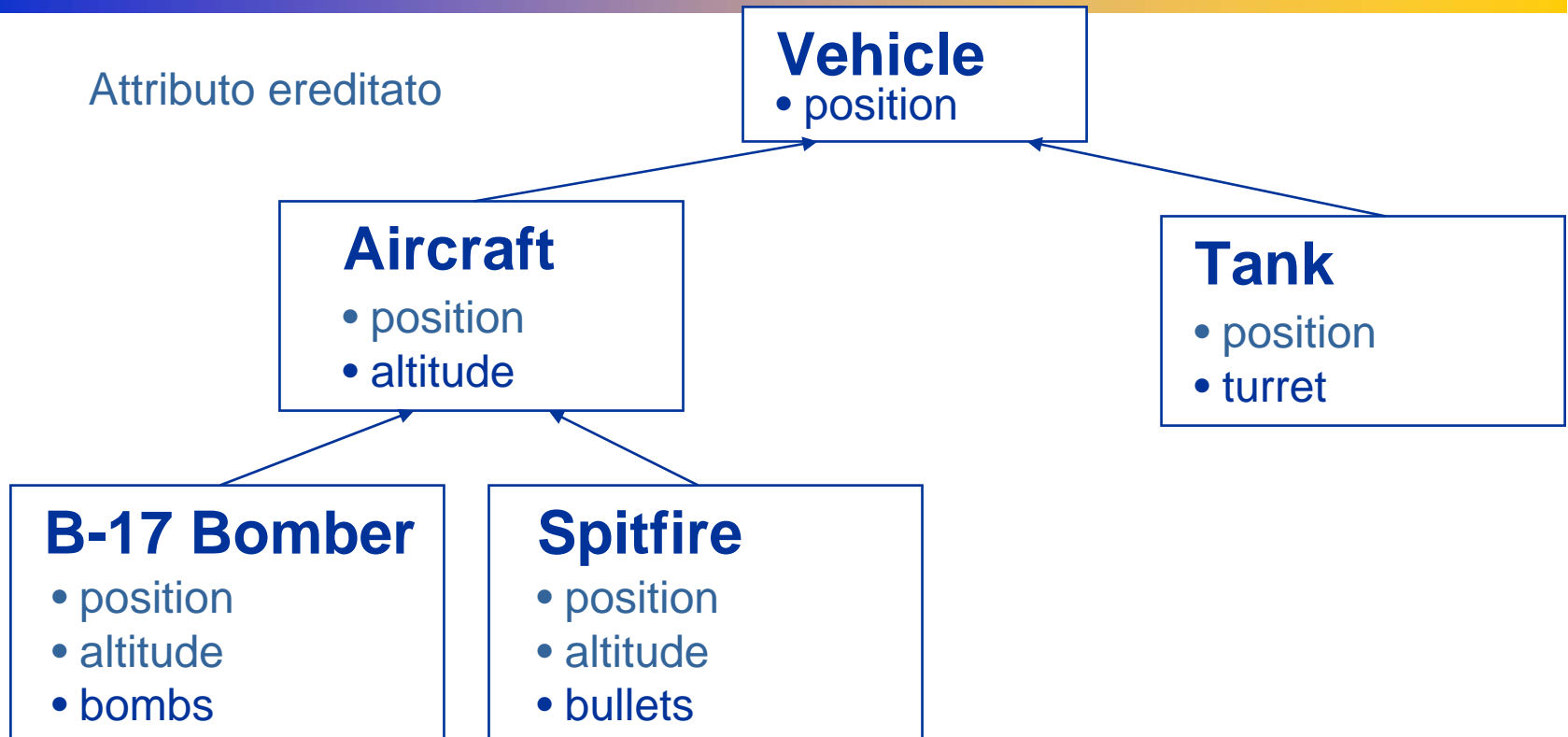
■ Dynamic Data Distribution

- Le proprietà sono dinamiche
- Esempio: il Data Distribution Management in HLA
 - *Routing spaces*
 - *Sottoscrizione all'attributo posizione degli oggetti "tank" che compaiono in uno specifico settore dell'area di gioco*

Class-Based Data Distribution

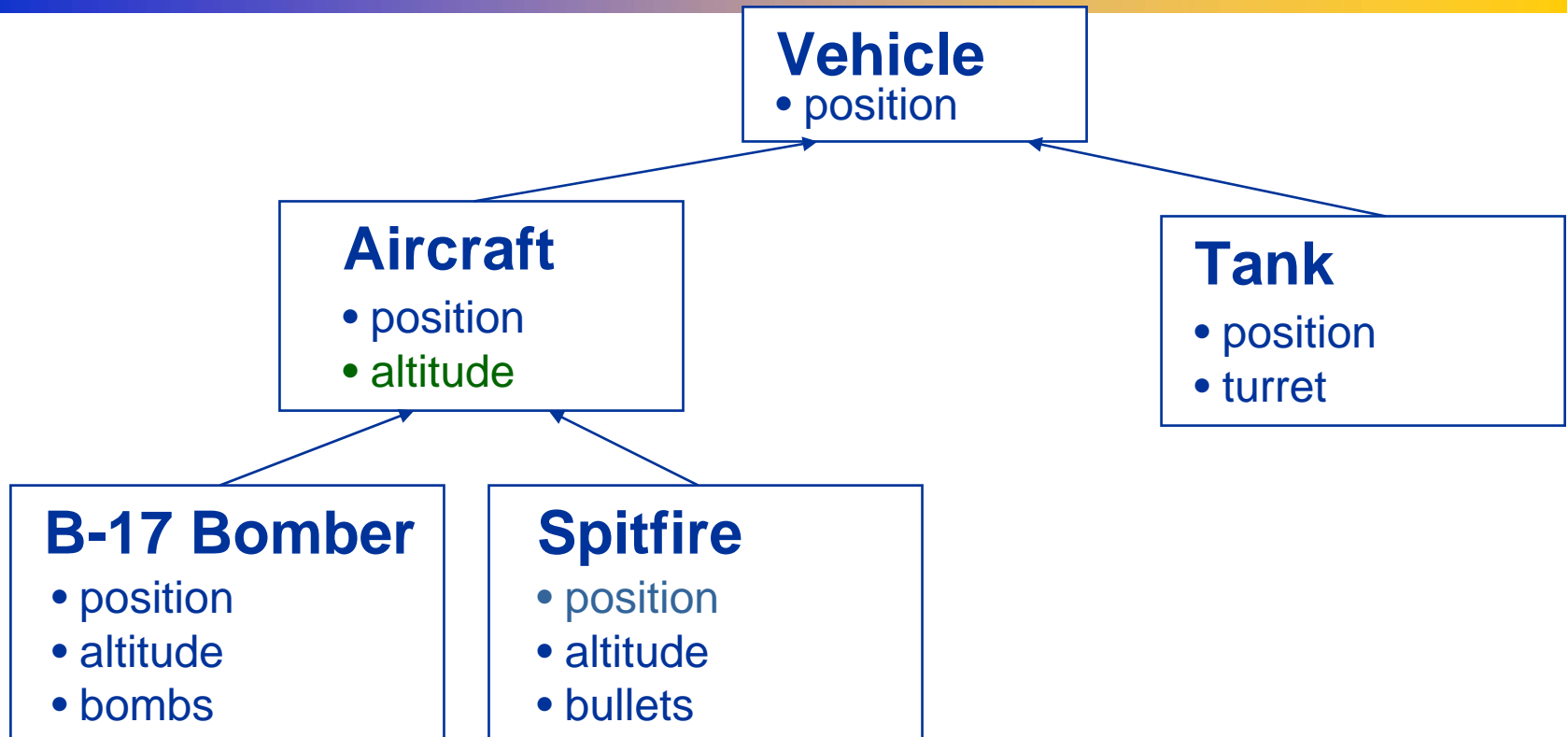
- Servizi di Declaration Management in HLA
- Il Federation Object Model (FOM) definisce una gerarchia tra classi di oggetti che descrive gli scambi di dati tra federati
 - Oggetti
 - Attributi
- Le Description Expressions e le Interest Expressions specificano quindi singoli punti all'interno della object class hierarchy

Class Hierarchy



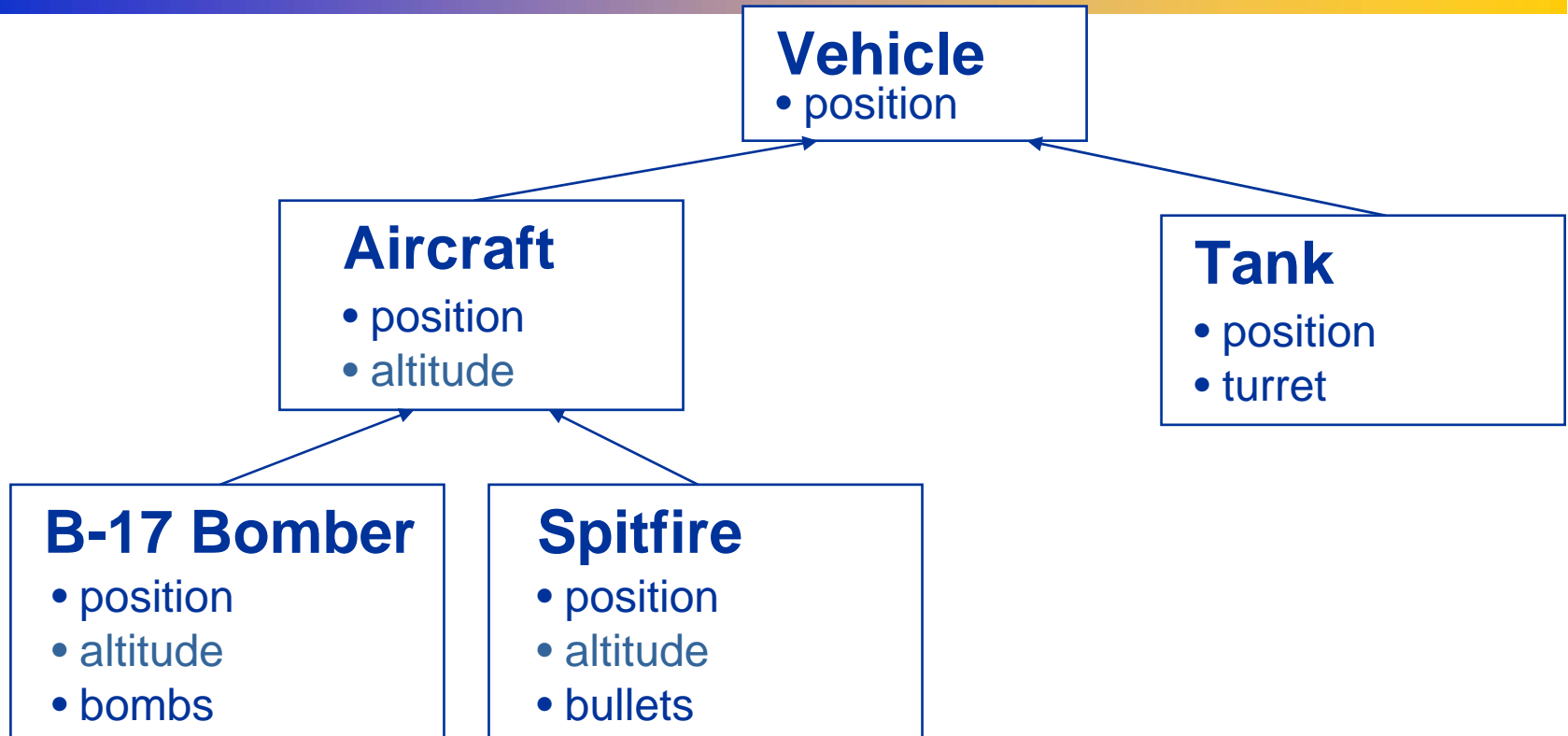
- Ogni classe eredita i suoi attributi dalla classe padre
- Name space: <class, attribute>
 - <Vehicle,position>, <Aircraft,position>, <Aircraft,altitude> ,
 - <Tank,position>, <Tank,turret>, <B-17,position>, <B-17,altitude> ,
 - <B17,bombs>, <Spitfire,position>, <Spitfire,altitude>, <Spitfire, bullets>

Description Expressions



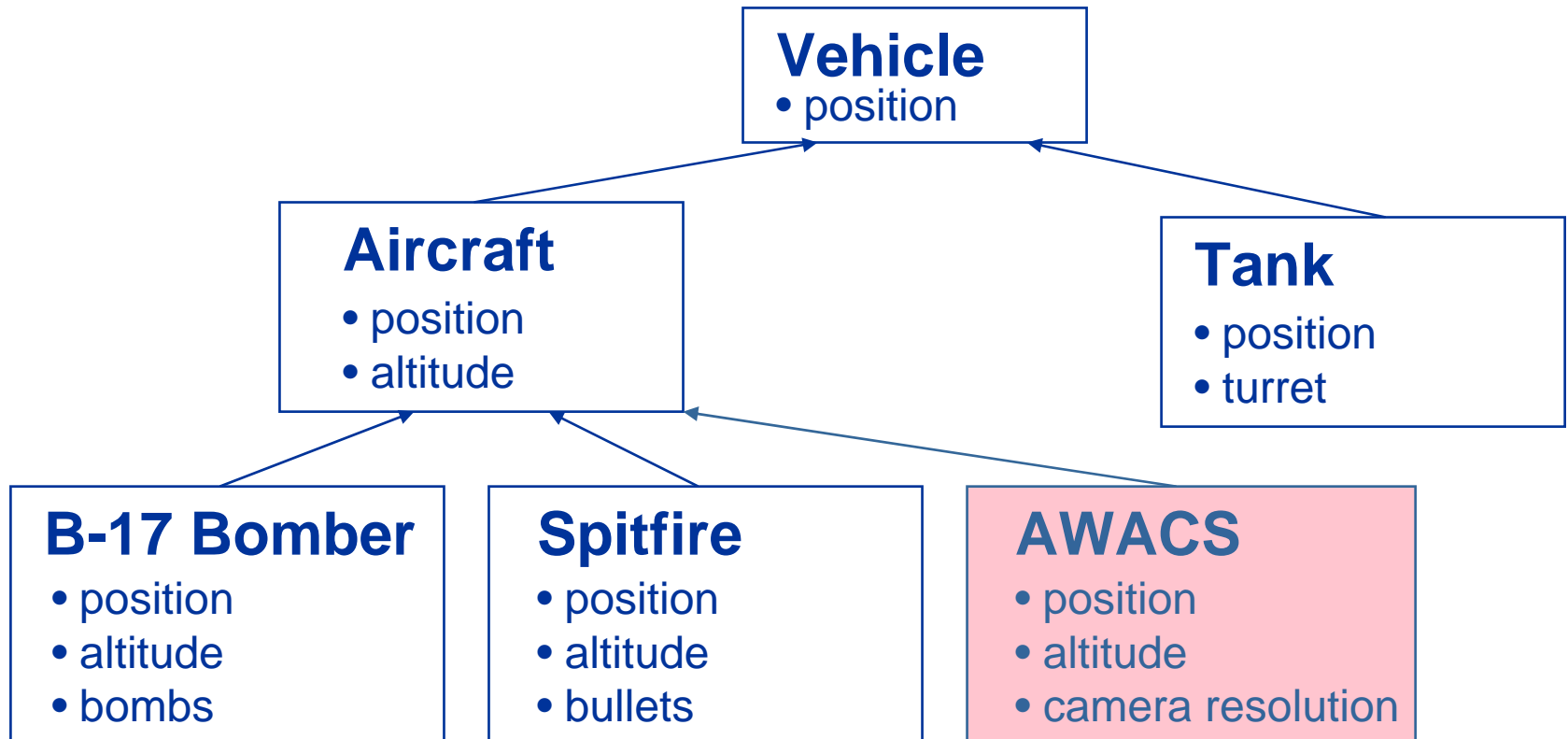
- Il servizio di aggiornamento degli attributi segnala un update
- Description expression: modifica attributo di un'istanza
 - Esempio: <Spitfire, position> oppure <Aircraft, altitude>

Interest Expressions



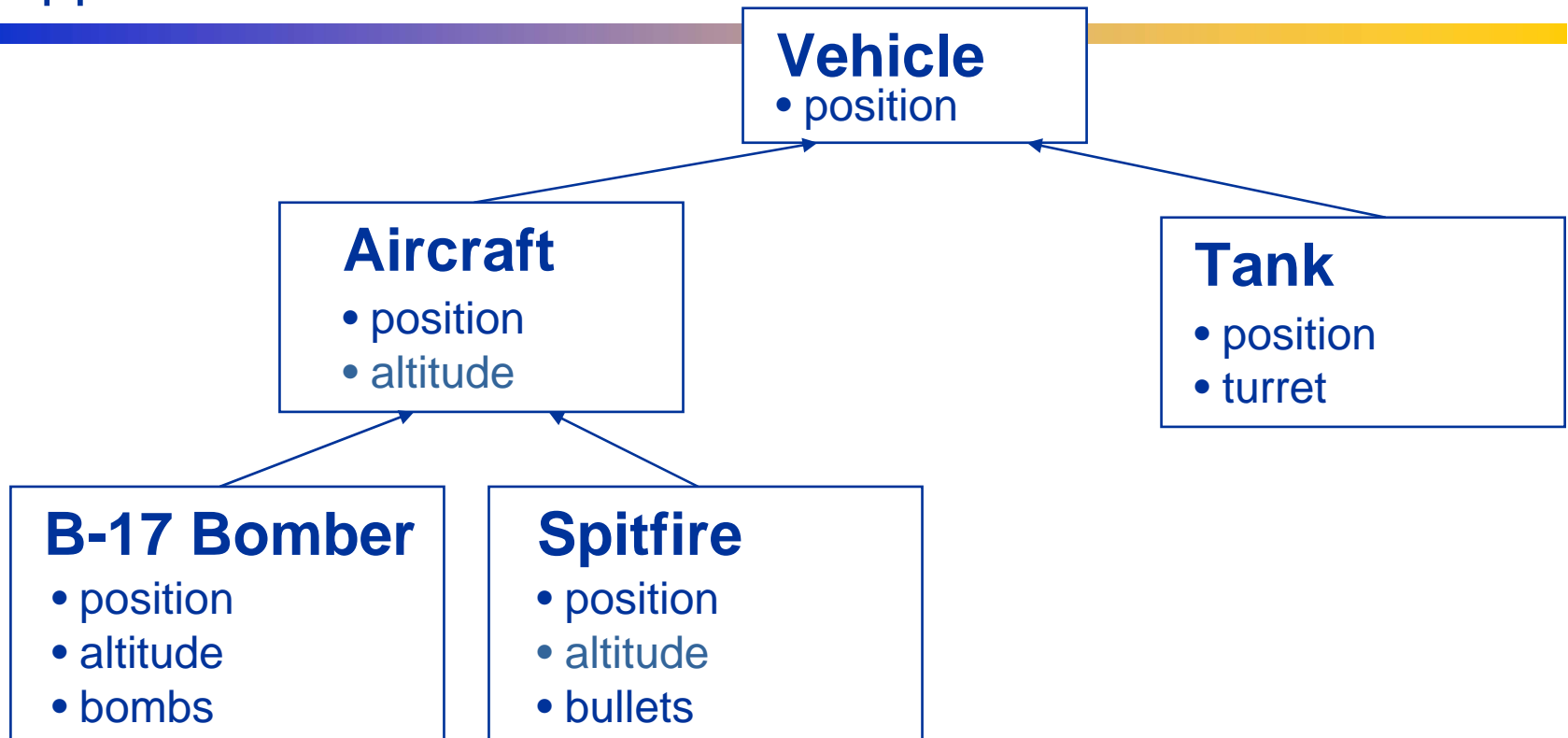
- Meccanismo di subscription [class, attribute(s)]
- Interest expression: sottoscrizione relativa sul sub-tree
 - Subscribe (Aircraft, altitude): riceve le update relative a tutti gli aerei!
 - <Aircraft,altitude>, <B-17 Bomber, altitude>, <Spitfire, altitude>

Vantaggi del meccanismo



Nuove classi possono essere aggiunte senza che sia necessaria alcuna modifica della gerarchia esistente (e della relativa implementazione dei simulatori)

Approccio alternativo



- Interest Expression: i subscribe operano su singole classi
 - `<Aircraft, altitude>`
- Description Expression: è necessario "risalire" ad Aircraft per trovare la definizione dell'attributo
 - Update (Spitfire, altitude): `<Spitfire,altitude>`, `<Aircraft,altitude>`

Conclusioni

- Per evitare i broadcast è necessario un sistema di Data Distribution
- Concetti fondamentali:
 - Name space
 - Interest Expressions (specifica informazioni ritenute interessanti dal simulatore = che vuole ricevere)
 - Description Expression (descrive le informazioni offerte per l'aggiornamento)
- Il compito del Runtime (RTI) è quello di determinare l'intersezione tra Interest Expression e Data Description Expression così da poter propagare correttamente le informazioni (solo quelle necessarie)

Simulazione Parallela e Distribuita

Data Distribution Management (DDM)

Prima parte



Gabriele D'Angelo

gda@cs.unibo.it

<http://www.cs.unibo.it/~gdangelo>

Dipartimento di Scienze dell'Informazione
Università degli Studi di Bologna