ALMA MATER STUDIORUM
UNIVERSITÀ DEGLI STUDI DI BOLOGNA

II facoltà di Ingegneria - Cesena
Ingegneria Informatica

Artifacts nel meta-modello A&A per sistemi multi agente: il framework CArtAgO

Sistemi Multi-Agente L-S

Michael Gattavecchia
0000362269

prof. Andrea Omicini

ANNO ACCADEMICO 2009/10

Indice

Introduzione

La ricerca nell'ambito della programmazione e dei sistemi multi-agente ha storicamente seguito una strada focalizzata all'analisi dei singoli agenti, in quanto principali protagonisti dei sistemi in oggetto. Ciò ha portato all'approccio a problematiche di sistema secondo prospettive particolarmente soggettive e saldamente fondate sulle capacità computazionali e comunicative degli agenti stessi, determinando sicuramente ottime caratteristiche di uniformità all'interno del sistema stesso. Un approccio di questo tipo dimostra però rilevanti carenze al momento di introdurre nel sistema meccanismi più articolati di coordinazione, in particolare si è notato come sistemi di questa natura non permettano un agevole scaling.

La complessità viene infatti interamente indirizzata verso gli agenti le cui specifiche tendono a crescere in modo molto rilevante, allontanando di fatto il sistema dal modello di riferimento e diminuento sensibilmente il livello di autonomia dello stesso, che si vorrebbe invece massimo.

Al fine di porre rimedio a queste fondamentali limitazioni è stato introdotto e sviluppato il concetto di *artifact*, che oltre a permettere una più efficiente rappresentazione dell'ambiente operativo in cui gli ageni svolgono le proprie attività, permette di sollevare quest'ultimi da gran parte delle responsabilità comunicative e computazionali necessarie.

Successive trattazioni sull'argomento [1] [2] suggeriscono l'introduzione degli artifact come astrazione di primo livello nella modellazione dei *workspace* all'interno dei quali gli agenti svolgono le proprie attività, arrivando poi [3] alla definizione di un nuovo meta-modello di riferimento, all'interno del quale viene riconosciuta agli artefatti il fondamentale ruolo di supporto alla coordinazione ed alle dinamiche di comunicazione necessarie.

Questo nuovo meta modello, chiamato A&A (Agents and Artifacts), diviene riferimentio per lo sviluppo di piattaforme sulle quali andare a definire sistemi complessi, all'interno dei quali agenti dotati di capacità cognitive possono porre il proprio focus sulle operazioni di reale interesse al fine del raggiungimento dei rispettivi goal interni, vedendo semplificate le operazioni relative alle dinamiche di coordinazione e comunicazione necessarie.

Il progetto CArtAgO nasce con lo scopo di fornire un supporto completo allo sviluppo ed all'esecuzione di ambienti artifact-based per sistemi multi-agente, strutturati in open workspace eventualmente distribuiti attraverso la rete ed accessibili ad agenti sviluppati attraverso molteplici modelli e piattaforme, grazie alla sua ortogonalità rispetto alle specifiche realtà applicative.

In questo approfondimento si tratterà dapprima il tema degli artifact, partendo dalle origini dell'astrazione fino ad arrivare alla formalizzazione del metamodello A&A, e si passerà quindi alla particolarità di CArtAgO come strumento per l'applicazione pratica del modello a sistemi multi-agente reali.

Capitolo 1

Artifacts

Analizzando gli schemi concettuali di riferimento, i sistemi ad agenti possono essere generalmente intesi come *goal-governed*: in cui agenti con capacità cognitive hanno la possibilità di rappresentare esplicitamente i propri obiettivi (goal) ed agire in funzione del loro completamento, e *goal-oriented*: in cui gli agenti sono concepiti e creati al fine di raggiungere un obiettivo interno anche non esplicitamente definito, senza ulteriori responsabilità cognitive ad essi delegate. In entrambi i casi si ha a che fare con goal *interni*, esistono poi obiettivi relazionati al contesto sociale o ambientale nel quale gli agenti sono posti: tali goal sono detti *esterni* e, ad esempio, nei sistemi di tipo goal-governed vengono raggiunti attraverso un triggering dinamico dei goal interni.

Esistono poi altre entità all'interno dei sistemi che, senza manifestare caratteristiche di pro-attività, possono venire *usate* dagli agenti al fine di raggiungere i loro obiettivi interni. Ci si riferisce a queste entità con il termine *artefatti* e, dualmente a quanto avviene per i termini goal e task nei confronti degli agenti, possono essere facilmente associati ai concetti di *uso* e *funzione*. Per meglio comprendere il ruolo degli artefatti all'interno del sistema è possibile ricercare analogie tra sistemi multi-agente e la nostra società, in cui si ha una buona corrispondenza logica tra il ruolo degli essere umani e gli agenti pro-attivi, e tra gli strumenti che gli uomini utilizzano per lo svolgimento delle proprie attività e gli artefatti, appunto.

1.1 Background multi-disciplinare

La nozione di sistema complesso attraversa i domini specifici di diverse discipline, tra cui la fisica, la biologia, l'economia, la sociologia ed i sistemi computazionali. Ciò dipende dal fatto che nell'analisi di sistemi, in contesti anche molto diversi, all'aumentare della complessità si assiste allo sviluppo di pattern spesso molto simili tra loro.

Riferendosi ai sistemi multi-agente diventa quindi molto utile andare a riferisi ad ambiti in cui sia possibile trovare casistiche analoghe a quelle oggetto di studio.

1.1.1 Activity Theory

La *Activity Theory* (AT), nata nell'abito della "Soviet Psychology", è una meta-teoria psicologica ad oggi largamente applicata in campo informatico: come ad esempio nel Computer Supported Cooperative Work (CSCW) e nella Human/Computer Interaction (HCI). Si tratta di una infrastruttura molto generica, orientata alla descrizione delle attività umane, che pone il concetto di *activity* come unità di analisi primaria.

Secondo questa teoria, nessuna attività svolta da uno o più partecipanti può essere compresa senza prendere in considerazione gli *strumenti* che rendono tali attività possibili o che svolgono una funzione di mediazione durante l'interazinoe tra gli attori coinvolti e l'ambiente nel quale sono posti. Gli stumenti, o artefatti, possono presentare una connotazione fisica (come ad esempio scaffali, porte, dispositivvi elettronici, lavagne, etc.), una connotazione cognitiva (come procedure operative, algoritmi, script, linguaggi, etc.) o eventualmente entrambe, come nel caso di manuali operativi o calcolatori.

In quanto strumenti di mediazione, gli artefatti hanno funzione *abilitante*, espandendo le potenzialità dell'utilizzatore attraverso l'interazione con l'ambiente, e *vincolante*, permettendo un'interazione coerente ad una specifica interfaccia o secondo un protocollo predefinito.

La Activity Theory si concentra su attività sociali (collaborative), modellandole secondo una struttura a tre strati: esse possono infatti essere di tipo *co-ordinative*, *co-operative* o *co-costruttive*.

- Gli aspetti *co-ordinativi* delle attività comprendono l'usuale schema di interazione all'interno del sistema, dove sia il significato che l'obiettivo relativi alle operazioni svolte sono dati e costanti nel tempo.

- Gli aspetti *co-operativi* delle attività si riferistono alle interazioni in cui gli attori si affidano ad un artefatto comune, che usano e condividono: in questo caso l'obiettivo dell'attività è condiviso e prestabilito, mentre il significato puntuale relativo all'operazione specifica non è ancora stato definito.

- Gli aspetti *co-costruttivi* delle attività, infine, concernono la ridefinizione dell'organizzazione interna degli attori e relativa alla coordinazione tra essi, in funzione degli strumenti condivisi. In questo caso significato puntuale delle azioni e obiettivi non sono predefiniti, ma vanno appunto costruiti tenendo conto dei vari contributi individuali.

La Activity Theory promuove dunque il concetto di *artefatto di coordinazione* per identificare quelle entità preposte alla mediazione delle interazioni, definendo in sostanza una sorta di generalizzazione del concetto di medium di coordinazione visti nell'ambito dei sistemi concorrenti e dell'AI.

1.1.2 Distribuited Cognition

Una visione simile è proposta dalla Distribuited Cognition che, nell'ambito delle scienze cognitive, suggerisce una rappresentazione della cognizione e della conoscenza umana non solo limitata agli individui, ma distribuita anche tra strumenti ed artefatti presenti nell'ambiente che incapsulano informazioni.

In questo caso gli artefatti, detti cognitivi, sono definiti[1] come "quegli strumenti artificiali che conservano, visualizzano o processano informazioni per fornirne una rappresentazione funzionale e che influenzano le prestazioni cognitive degli umani", essi sono prodotto dell'ingegneria umana e, oltre ad amplificare le abilità cognitive degli utilizzatori, modificano significativamente la dinamica secondo cui le attivià vengono svolte.

I sistemi, in questo ambito, possono essere osservati da due differenti punti di vista, uno *di sistema*, che considera attori ed artefatti nella loro totalità per la comprensione delle attività svolte e delle azioni portate a termine, ed uno *personale*, in cui si pone l'attenzione sugli attori per comprendere come gli artefatti influiscano sulle specifiche attività e modalità di interazione.

È importante tenere conto che lo scopo di una qualsivoglia attività non è semplicemente quella di modificare l'ambiente in modo da avvicinarsi all'obiettivo prefissato. Molte azioni hanno infatti, nel breve termine, lo scopo di alleggerie il carico complessivo sull'attore evitando sprechi di attenzione, memoria e computazione: come ad esempio accade quando si appunta un numero di telefono, che solo in un secondo, remoto, momento si vorrà richiamare. Tutto ciò acquista maggiore senso nell'assunzione che attori ed attività siano, a prescindere dal constesto specifico, *situati*, ovvero che sia presente un forte accoppiamento tra attori ed ambiente, a conferma dell'importanza dell'influenza sulle dinamiche comportamentali da parte degli artefatti.

1.1.3 Sociologia

In ambito sociologico viene fatta una prima distinzione tra entità contraddistinte da un *goal* interno, implicito o meno, ed entità prive di tale caratterizzazione. Riguardo quest'ultime, viene fatto notare come nella maggior parte

[1] Norman, D.A (1991). Cognitive Artifacts. In J. M. Carrol (Ed.), *Designing Interaction: Psychology at the human-computer interface*, Cambridge Series On Human-Computer Interaction (pp. 17-38). New York: Cambridge University Press

dei casi esse siano associabili al concetto di *uso*: possono cioè essere utilizzate dagli attori del sistema al fine di raggiungere i goal corrispondenti. Durante il design e la costruzione degli artefatti, viene ad essi associata una *funzione* che ne indirizza l'uso da parte degli attori e ne influenza l'interfaccia ed il comportamento.

Il concetto di *destinazione* è legato (ma non corrispondente) al concetto di funzione: la destinazione di un artefatto, più che all'entità responsabile della sua creazione, è legata all'utilizzatore finale dell'artefatto, che ne valutà l'utilità e le potenzialità in funzione delle proprie necessità particolari. Secondo questo schema, ad un artefatto può essere associata una destinazione anche diversa dalla funzione per la quale è stato progettato.

Esistono due tipologie di *goal esterni* che gli attori associano agli artefatti:

- *use-value goal*, obiettivo secondo il quale l'artefatto ha le potenzialità di permettere all'utilizzatore di raggiungere il proprio obiettivo, e che lo mette in condizione di operare la *selezione* dell'artefatto più opportuno.
- *use goal*, che corrisponde direttamente al goal interno dell'utente, e determina la modalità puntuale di utilizzazione dell'artefatto stesso.

In questo contesto un artefatto viene dunque creato per uno specifico scopo, che ne influenza la selezione da parte degli utenti, i quali possono però farne l'uso che ritengono più efficiente in funzione dei propri goal interni.

1.1.4 Computer Supported Cooperative Work (CSCW)

Il CSCW segue, più o meno esplicitamente, due principali strategie apparentemente divergenti: la prima tende a promuovere l'automatizzazione della coordinazione tra gli utenti, la seconda a favorire la flessibitlià delle interazioni all'interno del sistema. La convergenza tra i due approcci può essere ottenuta attraverso la conoscenza reciproca tra gli attori del sistema ed il ricorso ad artefatti di coordinazione specifici. In particolare è possibile ottenere il grado di automazione desiderato delegando ad artefatti di coordinazione appositi il ruolo di regolatori delle comunicazioni: la definizione di una modalità di comunicazione comune mette in condizione gli attori di interagire in modo efficace senza porre limiti alla flessibilità delle interazioni stesse. Questa flessibilità verrà poi abilitata attraverso la conoscenza reciproca, garantita dalla costruzione di workspace distribuiti, che concorreranno alla costruzione di un ambiente di lavoro comune (common field of work).

1.1.5 Antropologia ed etologia

L'antropologia occidentale ha da sempre associato la presenza di linguaggi simbolici allo sviluppo dell'intelligenza umana, andando nel tempo ad assegnare una sempre minore importanza al rapporto tra l'intelligenza stessa e gli strumenti costruiti ed utilizzati per il compimento delle principali attività.

Solo recentemente[2], dopo decenni di ricerca in differenti ambiti, si è iniziato ad affrontare l'argomento dello sviluppo delle capacità cognitive dell'uomo, del linguaggio e degli strumenti da esso utilizzati come un'unica, coerente problematica. Analogamente, nel regno animale, viene spesso associato il concetto di intelligenza alla capacità di affrontare problemi non noti, risolvibili esclusivamente con il ricorso a strumenti appositi.

Uno strumento rivela la conoscenza di sè e/o del mondo nel quale ci si trova, nel momento in cui esso sia costruito per un fine, sia conservato per un eventuale riutilizzo (anche periodico) o sia utilizzato per la costruzione di altri strumenti.

Risulta molto rilevante riportare come una variante, basata sugli strumenti, del test di Turing sia stata recentemente proposta in ambito psicologico per la valutazione dell'intelligenza in termini di abilità nell'utilizzo di strumenti, appunto, e per questo chiamata "Tooling Test for Intelligence".

1.2 Modello

Dalle considerazioni multi-disciplinari precedenti è possibile estrarre un primo modello dell'astrazione di *artefatto* e formularne una definizoine. Un artefatto può essere definito come:

> "un dispositivo all'interno di un ambiente popolato da agenti, progettato per offrire o svolgere una qualche forma di servizio o funzione nei confronti degli agenti, individualmente o collettivamente, al fine di permettere loro di raggiungere i propri obiettivi e a supporto delle loro attività."

In figura 1.1 viene riportata una rappresentazione astratta di artefatto, caratterizzata da quattro elementi principali: *l'interfaccia di utilizzo (UI)*, le *istruzioni operative (OI)*, la *funzione* e la *stuttura e comportamento*.

Usage Interface (UI) è l'insieme delle *operazioni* che gli agenti possono invocare per utilizzare l'artefatto e sfruttare le potenzialità da esso offerte. All'invocazione di un'operazione da parte di un agente può corrispondere, in un momento futuro non predeterminato, l'occorrenza di un evento di

[2]Gibson, K. R. & Ingold, T. (Eds.), (1993) *"Tools, language and cognition in human evolution"*, Cambridge University Press.

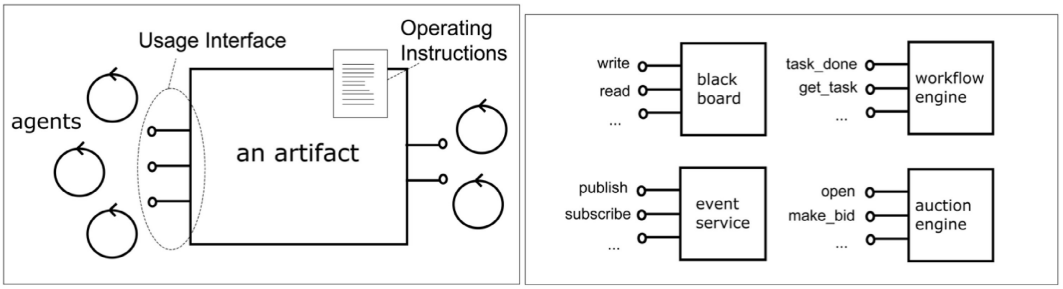

Figura 1.1: Una rappresentazione astratta di artefatto ed alcune specifiche istanze d'esempio.

un qualche tipo, tipicamente identificato dalla comunicazione di informazioni relative all'esito dell'operazione stessa. Questi eventi possono essere percepiti dagli agenti come esterni (percezioni).

Operating Instructions (OI) sono le descrizioni di 'come' utilizzare l'artefatto per far uso delle sue potenzialità. Le istruzioni operative descrivono i possibili *protocolli di utilizzo* dell'artefatto e possono eventualmente comprendere una descrizione semantica delle operazioni da svolgere, favorendo così un eventuale uso cognitivo dello stesso.

Function è lo scopo per il quale l'artefatto è stato progettato ed implementato da progettisti e programmatori all'atto della sua definizione.

Structure and behaviour infine, concerne gli aspetti architetturali interni, anch'essi dipendenti dalla specifica implementazione funzionale allo scopo designato. Tali aspetti sono tipicamente ignoti a quelli che saranno gli utilizzatori finali dell'artefatto.

A differenza degli agenti, gli artefatti non sono intesi per manifestare comportamenti pro-attivi. Oltre alle caratteristiche elencate in precedenza sono apprezzabili, in funzione del fine per il quale gli artefatti sono intesi, alcune ulteriori funzionalità:

inspectability and controllability la possibilità, da parte degli agenti utilizzatori, di osservare e controllare a run-time lo stato dell'artefatto ed il suo comportamento, oltre alla possibilità di svolgere attività di diagnostica, debug e testing.

malleability possibilità di modificare o adattare la funzione dell'artefatto a run-time, in corrispondenza di nuovi requisito o eventi non previsti.

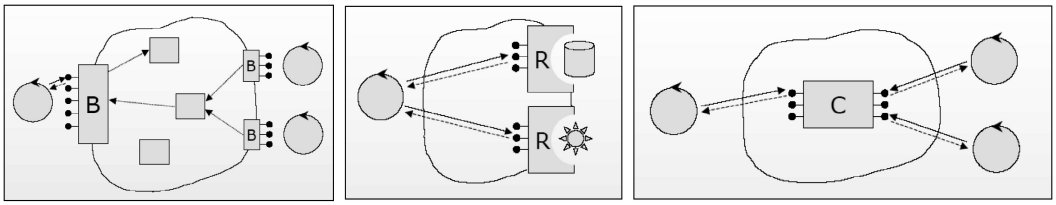

Figura 1.2: Tipologie ricorrenti di artefatti: boundary artifacts (B), resource artifacts (R), coordination artifacts (C).

linkability possibilità di raggruppare più artefatti a run-time, permettendo un buono scaling della complessità ed un buon riuso dinamico delle risorse presenti.

Oltre alla differenza comportamentale nei confrondi degli agenti, gli artefatti sono contraddistinti da una connotazoine spaziale, ovvero sono accoppiati con quella che è la topologia del sistema multi-agente all'interno del quale sono posti. Ciò suggerisce come sia possibile avere artefatti logicamente e/o fisicamente distribuiti in rete.

Da questo modello concettuale è possibile identificare tre principali caratteristiche per il relazionamento tra artefatti ed agenti: *(i)* utilizzo, *(ii)* selezione e *(iii)* costruzione e manipolazione. Tipicamente per utilizzare un artefatto è sufficiente che un agente ne conosca l'interfaccia di utilizzo, per la selezione sono fondamentali funzione e destinazione, mentre costruzione e manipolazione sono tendenzialmente caratteristiche relative all'implementazione specifica dell'artefatto.

1.3 Esempi

Per meglio comprendere i concetti esposti è utile analizzare alcune tipologie di artefatti spesso ricorrenti nei sistemi multi-agente, come visibile in figura 1.2. Gli esempi riportati di seguito, organizzati per funzione, non sono da intendere come rigorosi riferimenti per la creazione di artefatti.

Coordination artifacts si tratta di artefatti progettati al fine di abilitare la coordinazione tra agenti all'interno del sistema. Nell'ambito della computer science sono molti i meccanismi introdotti a questo fine, a partire da artefatti orientati alla sola comunicazione, in cui l'onere della coordinazione viene lasciato agli agenti utilizzatori, fino ad arrivare a rappresentazioni con specifiche funzioni legate al sincroniscmo, come scheduler o semafori. Il concetto di

artefatto di coordinazione è molto vicino a quello di *medium di coordinazione*, sviluppato nell'ambito dei sistemi concorrenti ma, mentre nella quasi totalità delle trattazioni presenti in letteratura ci si orienta alla coordinazione tra processi, nel caso degli artefatti sono stati sviluppate alcune caratteristiche mirate all'utilizzo cognitivo degli stessi, in particolare da parte di entità goal-governed.

Vale la pena di notare come gli artefatti di coordinazione rappresentino un approccio *ingegnerizzato* al problema della coordinazione, in quanto in essi la fase di definizione delle dinamiche di coordinazione viene effettuata prima della effettiva implementazione, potendo così affrontare le problematiche in modo ottimale. Tuttavia, le caratteristiche di dinamicità dei sistemi possono portare a casi in cui non sia possibile svolgere analisi preventive: in questi casi gli artefatti possono essere utilizzati al solo fine di abilitatori all'interazione, lasciando *emergere* le dinamiche di coordinazione, analogamente a quanto osservabile in molto sistemi biologici.

Boundary artifacts sono una caso particolare di artefatti organizzazionali, con particolari funzioni orientare ad organizzazione e sicurezza. Un artefatto di tipo boundary (BA) viene utilizzato per caratterizzare e controllare la presenza di un agente all'interno di uno specifico contesto. Ad esempio, in ambienti con politiche di tipo role-based, un BA incapsula la regolamentazione relativa ai ruoli assegnati ai vari agenti all'interno del sistema. In termini più generali, quindi, un BA può essere definito come entità che incapsula le regole ed i vincoli intercorrenti tra un agente e l'ambiente nel quale opera.

Resource artifacts artefatti progettati per la mediazione dell'accesso ad una specifica risorsa o per la diretta rappresentazone ed incapsulamento della risorsa stessa. Questa tipologia di artefatto è fondamentale in quanto porta al livello di astrazione dell'agente tutte le entità computazionali potenzialmente utili ad esso.

Seppure a volte modellare risorse utilizzando l'astrazione di agente risulti comodo, nella maggioranza dei casi si hanno risultati migliori utilizzando artifatti che, per definizione, risultano non pro-attivi ed in grado di fornire qualche sorta di funzione o servizio agli utilizzatori. È importante anche tenere conto che, dal punto di vista implementativo, gli artefatti sono generalmente caratterizzati da strutture molto più snelle rispetto agli agenti, e ciò può diventare un particolare di notevole interesse nel caso (frequente) in cui si debbano realizzare sistemi complessi.

Capitolo 2

Meta-modello A&A

2.1 Concetti di base

Facendo riferimento alle fondamenta concettuali discusse in precedenza, il meta-modello A&A è caratterizzato da tre astrazioni fondamentali:

Agenti, elementi pro-attivi che incapsulano l'esecuzione autonoma di svariate tipologie di attività all'interno di qualche tipo di ambiente.

Artefatti, componenti passivi del sistema, come risorse e supporti multimediali, appositamente costruiti, condivisi, modificati ed utilizzati dagli agenti a supporto delle proprie attività, sia a fini cooperativi che competitivi.

Workspaces, contenitori logici di agenti ed artefatti, utili per la definizione della topologia dell'ambiente e per il supporto alla definizione del concetto di località.

Vengono di seguito riportate alcune definizioni di agente, artefatto e sistema multi-agente presenti in [3], che non vogliono ridefinire i concetti di agente ed artefatto al livello delle loro attuali caratterizzazioni comuni o tecniche, ma sono piuttosto orientate all'esplicitazione del significato di tali concetti nell'ambito del meta-modello A&A.

Le definizioni che seguiranno adottano l'approccio lessicale secondo il quale le definizioni scientifiche sono espresse in termini di *Genus* e *Differentia*: in sostanza viene chiaramente definito il contesto che di fatto delimita il dominio dell'argomento (genus) e vengono quindi esplicitate la caratteristiche specifiche che si intende definire (differentia).

2.1.1 Definizioni

La caratteristica fondamentale riconosciuta dal meta-modello A&A per gli agenti è sicuramente l'*autonomia*.

Dal punto di vista computazionale, definire gli agenti autonomi significa assumere che essi incapsulino il *flusso di controllo*, che non devono cedere all'esterno a meno di casi particolari in cui sono essi a manifestare esplicitamente la volontà di farlo. Come conseguenza di questa specifica gli agenti non mettono a disposizione interfacce finalizzate al proprio controllo da parte di entità esterne, inoltre si può evincere come i sistemi multi-agente debbano essere visti come aggregazione di molteplici flussi di controllo distinti, in grado di interagire reciprocamente attraverso lo scambio di informazioni.

Definire gli agenti come entità autonome all'interno dei MAS significa assumere che in essi siano racchiusi i concetti di self-government, self-regulation e self-determination, che corrisponde all'assunzione implicita secondo cui essi incapsulino anche i criteri per la gestione del flusso di controllo, sostanzialmente rappresentabili dai *goal* citati in precedenza.

La pro-attività[1] degli agenti rende necessaria la specifica del concetto di *azione*, a sua volta dipendende da quello di *cambiamento* all'interno del sistema. Il vincolo sullo scambio di sole informazioni tra agenti porta poi a riferirsi a tali azioni come *speech act* o *communication act*.

A&A Agent

Un agente nel metamodello A&A è un'*entità computazionale autonoma*.

genus Gli agenti sono entità computazionali

differentia Gli agenti sono autonomi, in quanto essi incapsulano il flusso di controllo ed i criteri per la sua gestione

A&A Artifact

Un artefatto è un'*entità computazionale* intesa per essere *utilizzata* da parte degli agenti A&A.

genus Gli artefatti sono entità computazionali

differentia Gli artifatti vengono utilizzati dagli agenti

[1] sostanzialmente la capaticità di generare eventi, senza quindi limitarsi all'attesa passiva dell'occorrenza di questi alla quale far seguire una reazione.

La prima fondamentale differenza che è possibile notare è relativa alla presenza di autonomia, della quale gli artefatti sono privi in quanto intesi come fornitori di un servizio verso gli agenti. Ogni artefatto è infatti costruito con riferimento ad una *funzione* per la quale esso è inizialmente inteso, e per essere utilizzato offre agli agenti una serie di operazioni in grado di modificare il suo stato interno, esplicitate nella sua *interfaccia*.

A&A Multi Agent System

Un sistema multi-agente, nell'ambito del meta-modello A&A è un *sistema computazionale* formato da agenti ed artefatti.

genus I sistemi multi-agente sono sistemi computazionali

differentia I componenti principali dei sistemi multi-agente sono agenti ed artefatti

Questa descrizone non è esaustiva in quanto non tiene conto del concetto di *workspace*, che nel contesto di sistemi situati guadagna ulteriore rilevanza.

L'insieme delle interazioni possibili all'interno di un sistema multi-agente, nell'ambito del meta-modello A&A, è composto da quattro attività nelle quali possono essere coinvolti agenti e/o artefatti:

communication agenti *comunicano* con altri agenti

operation agenti *utilizzano* artefatti

composition artefatti si *connettono* con altri artefatti

presentation artefatti *esibiscono* la propria interfaccia agli agenti

Oltre a queste operazioni saranno presenti interazioni con l'environment del sistema, che dipenderanno dallo specifico livello di astrazione desiderato/ di riferimento.

2.2 Uso cognitivo degli artefatti

Con il riferimento al concetto di *uso* nella definizione di artefatto e del rapporto tra artefatti ed agenti, diventa necessario chiedersi se l'astrazione di artefatto sia sufficiente per l'abilitazione di comportamenti cognitivi da parte degli agenti.

Analizzando il modello proposto appare chiaro come gli artefatti possano influenzare in modo anche molto rilevante il comportamento degli agenti, che dovranno essere in grado di svolgere il delicato compito di *selezione* dell'artefatto più efficace relativamente al raggiungimento dei goal di riferimento.

2.2.1 Cognitional artifacts

Per mettere gli agenti in condizione di valutare i possibili esiti dipendenti dall'utilizzo di un artefatto è necessario incapsulare le corrispondenti caratteristiche strutturali e comportamentali all'interno dello stesso, in modo da mettere l'agente in condizione di acquisire tali informazioni attraverso una interrogazione apposita. La definizione di artefatto relativa al meta-modello A&A fornita in precedenza risulta insufficiente a tale scopo, rendendo così necessario andare a definire una nuova entità, chiamata *artefatto cognitivo*, che abilita l'utilizzo di criteri cognitivi all'interno delle dinamiche interazionali del sistema multi-agente.

A&A Cognitional Artifact

Un *artefatto* cognitivo nell'ambito del meta-modello A&A è un artefatto che permette agli agenti di fare di esso un uso di tipo cognitivo.

genus Gli artefatti cognitivi sono artefatti

differentia Gli artefatti cognitivi possono essere utilizzati secondo meccanismi cognitivi da parte degli agenti A&A

Con il termine "uso di tipo cognitivo" si intende un uso da parte degli agenti in cui il razionale nelle interazioni sia guidato dalle capacità cognitive degli stessi, che rende necessaria la presenza di particolari caratteristiche negli artefatti, analogamente a quanto esposto ai capitoli precedenti:

- Function Description
- Operating Instructions
- Usage Interace

2.2.2 Selezione di un Artifact attraverso la function description

Coerentemente al meta-modello A&A, un artefatto cognitivo deve essere contraddistinto da una *function description* tale da mettere gli agenti in condizione di valutare il tipo ed il grado di beneficio che potrebbero trarre dal suo utilizzo. Per gli agenti, in particolare, risulta importante il concetto di *use-value*, che funge da discriminante per la selezione dell'artefatto più adatto all'interno dell'insieme di quelli presenti nel workspace. I modi in cui la function description di un artefatto viene mostrata agli agenti possono essere molteplici, in funzione ad esempio della sintassi e della semantica del linguaggio utilizzato.

Queste ovvie considerazioni possono essere sicuramente estese alle operating instructions ed alla usage interface, in quanto la scelta della metologia di rappresentazione delle informazioni dipende da molteplici fattori, come lo scenario specifico, l'eventuale esistenza di un'ontologia comune per tutti gli agenti, il grado di apertura del sistema, etc.

2.2.3 Utilizzo degli Artifact tramite Operating Instructions

Una volta operata la *selezione* dell'artefatto più adatto, un agente con capacità cognitive, deve ora poter essere in grado di venire a conoscenza di *come* sfruttare nello specifico le potenzialità dell'artefatto. La descrizione delle operazioni è resa disponibile, analogamente a quanto detto per gli artefatti in generale, attraverso le *operating instructions*.

Le istruzioni operative devono, ovviamente, essere scritte in un linguaggio comprensibile agli agenti, e vanno a definire la semantica delle sequenze di operazioni ammissibili o richieste per l'utilizzo dell'artefatto. Il contributo apportato dalle operating instructions è l'assistenza agli agenti nel momento della creazione del proprio piano d'azione.

I linguaggi utilizzati per l'espressione delle istruzioni possono essere diversi a seconda delle caratteristiche degli agenti che si scelgono come target, ad esempio con linguaggi operazionali in cui vengono specificate pre e post condizioni relative alle operazioni è possibile mettere in condizione di utilizzare gli artefatti anche ad agenti non dotati di particolare intelligenza, in alternativa è possibile l'uso di linguaggi maggiormente dichiarativi o logic-based.

2.2.4 Operazioni sugli Artifact: la Usage Interface

Le operative instructions mettono gli agenti in condizione di decidere come operare sull'artefatto, e per la messa in pratica di tali intenti è infine necessario fornire una *usage interface*, che rappresenta l'insieme delle possibili operazioni messe a disposizione dall'artefatto.

Per ogni operazione viene fornita una descrizione di come essa debba essere invocata e di come il prodotto in uscita possa essere percepito dagli agenti. Ogni operazione è tipicamente descritta attraverso: *(i)* nome, *(ii)* argomenti (nomi e tipi) e *(iii)* output.

Un agente può eseguire un'azione su di un artefatto specificando il nome dell'operazione e fornendo i parametri in ingressi necessari. Il risultato dell'operazione comprende gli eventi generati dall'artefatto, che possono essere percepiti dall'agente. Tali eventi possono segnalare *(i)* quando l'operazione viene di fatto avviata , *(ii)* se l'operazione viene completata con successo, e

l'esito specifico della stessa, *(iii)* se e per quale motivo si è verificato un fallimento, e *(iv)* se l'artefatto ha raggiunto uno stato che può essere di interesse per l'agente.

La usage interface può eventualmente essere dinamica, ovvero mostrare particolari operazioni solo in funzione dello stato attuale dell'artefatto, e possono in oltre essere raggruppate in sottoinsiemi logici accomunati dalla tipologia delle operazioni corrispondenti.

2.2.5 Proprietà degli artefatti cognitivi

Le tre carratteristiche principali viste per gli artefatti congnitivi (i.e. function description, operating instructions, usage interface) supportano pienamente l'utilizzo da parte di agenti cognitivi durante la selezione, il planning operativo e l'interazione. Tuttavia, in molti scenari è possibile abilitare l'uso cognitivo degli artefatti, secondo diversi gradi di complessità, anche in presenza di un sottoinsieme delle suddette caratteristiche.

Ad esempio, nel caso in cui non fosse disponibile una function description, un agente potrebbe essere comunque in grado di dedurre le caratteristiche funzionali di un artefatto attraverso test di tipo trial-and-error, o attraverso l'osservazione del comportamento di altri agenti presenti nel sistema. Analogamente potrebbero venire risolte le mancanze di usage interface o operating instructions.

Ciò che realmente caratterizza gli artefatti cognitivi (tenendo conto che le caratteristiche in essi presenti potrebbero essere implicitamente definite anche in artefatti non cognitivi) è il fatto che tali caratteristiche vengono esplicitamente progettate al momento del design dell'artefatto stesso. Ciò permette di mettere gli agenti cognitivi del sistema in condizione di ispezionare gli artefatti presenti nell'ambiente ed elaborare decisioni relative al loro eventuale utilizzo, permettendo così la creazione di sistemi multi-agente realmente aperti in cui è possibile ambire alla realizzazione di entità davvero autonome.

Ovviamente, lo scenario descritto è abbastanza differente rispetto ai tipici sistemi multi-agente presenti in letteratura: l'esistenza di artefatti, ed in particolare di artefatti cognitivi, rende necessario riflettere sul concetto di "intelligenza degli agenti", tenendo conto delle capacità comunicative degli agenti relativamente ad un linguaggio simbolico, ma anche dell'abilità degli stessi di ricorrere all'utilizzo di artefatti per riuscire e completare i propri goal.

Capitolo 3

Il framework CArtAgO

L'astrazione di artefatto discussa ai capitoli precedenti rappresenta il cuore del framework concettuale CArtAgO (Common "Artifacts for Agents" Open framework), con particolare riferimento al meta-modello A&A. Generalmente è possibile intendere gli artefatti come entità computazionali passive e *function-oriented*, esplicitamente progettati allo scopo di assolvere ad un qualche tipo di *funzione*, e di essere a tale scopo *utilizzati* dagli agenti.

Questo scenario influenza direttamente i concetti di interazione all'interno del sistema multi-agente, in particolare gli agenti sviluppano le proprie azioni attraverso *(i)* l'elaborazione, *(ii)* la comunicazione reciproca, e attraverso *(iii)* l'utilizzo e l'eventuale costruzione di artefati condivisi.

Il framework CArtAgO permette la prototipazione di sistemi multi-agente dotati di ambienti operativi artifact-based, mettendo a disposizione *(i)* l'API per la definizione degli artefatti, *(ii)* l'API per permettere agli agenti di interagire con gli artefatti presenti nell'ambiente, e *(iii)* un ambiente run-time in grado rappresentare e gestire dinamicamente un ambiente di lavoro. CArtAgO non introduce ulteriori specifici modelli o architetture per la rappresentazione di agenti ed ambienti, esso è invece inteso per offrire supporto alle piattaforme ad agenti già esistenti, rendendone possibile l'eventuale interoperabilità.

3.1 Architettura e modello astratto

Si prendono di seguito in considerazione la struttura di base e gli elementi principali di CArtAgO, con riferimento allo schema architetturale astratto mostrato in figura 3.1.

L'architettura astratta di CArtAgO è composta da tre principali elementi: *(i) agent bodies*, entità che rendono possibile l'attuazione della località spaziale (situatedness) all'interno dell'ambiente di lavoro, *(ii) artefatti*, ele-

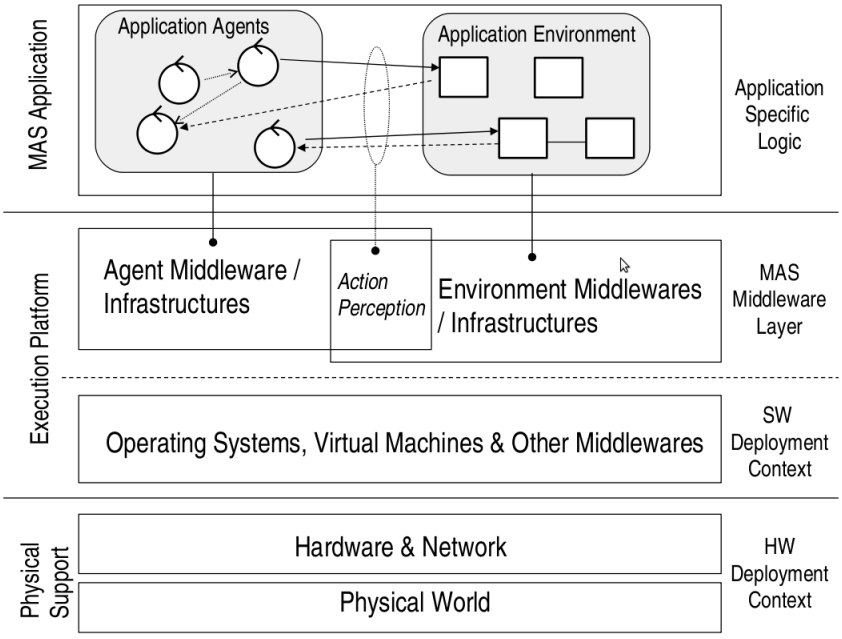

Figura 3.1: Schema architetturale generale di riferimento.

menti fondanti della struttura dell'ambiente, e *(iii) workspaces*, contenitori logici di artefatti, utili per la definizione della topologia degli ambienti.

3.1.1 Agent bodies

Gli agent bodies sono l'elemento in grado di attuare l'accoppiamento tra l'agente ed un ambiente di lavoro CArtAgO. Per ogni agente intenzionato ad operare all'interno dell'ambiente CArtAgO viene creato un body dedicato, contenente gli *effectors* in grado di compiere azioni, ed un set dinamico di *sensors* per la percezione degli stimoli provenienti dall'ambiente. Il body è inteso per essere controllato dall'agente, che di fatto ricopre il ruolo di "pilota", interagendo attraverso l'interfaccia di controllo esposta.

L'interazione del body con l'ambiente avviene attraverso l'esecuzione di *azioni*, appositamente create per la costruzione, selezione e l'uso di artefatti, o per la percezione di *eventi osservabili* generati da quest'ultimi.

Differentemente dagli schemi usualmente riscontrati, in questo caso la percezione è modellata come un atto volontario, a cui ci si riferisce con il termine *sensing*. In particolare, gli eventi osservabili all'interno dell'ambiente e generati dagli artefatti sono immagazzinati dai *sensori*, facenti parte dell'agent body. Un

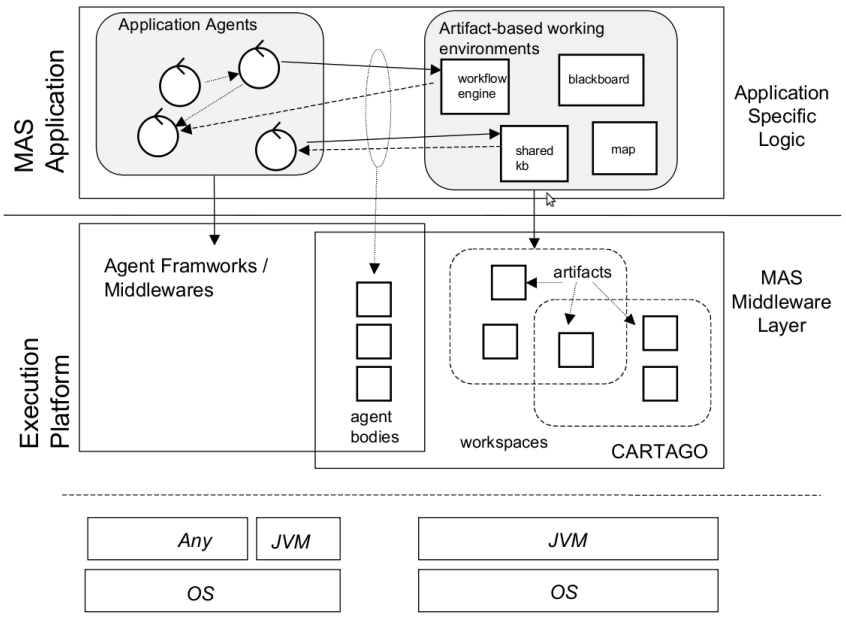

Figura 3.2: Schema architetturale con supporto al framework CArtAgO

agente può inoltre (dis)connettere dinamicamente sensori di varie tipologie al proprio corpo in funzione delle necessità.

3.1.2 Artefatti

Gli artefati sono i blocchi fondamentali gestiti da CArtAgO, ognuno di essi possiede un *nome* logico specificato al momento dell'istanziamento, ed un *id* univoco ad esso contestualmente assegnato dal framework. Il nome completo (*full name*) di un artefatto include il nome del (o degli) workspace all'interno del quale esso è localizzato. Il fatto che un artefatto possa essere associato a più workspace contemporaneamente fa sì che sia possibile riferirsi ad esso in più modi, equivalenti.

Interfaccia d'uso ed eventi osservabili

Il meccanismo di interazione tra agenti ed artefatti è fondato sui concetti di uso ed osservazione. Gli agenti possono utilizzare gli artefatti invocando l'esecuzione di una delle operazioni presenti nella rispettiva *usage interface*. L'esecuzione di un'operazione in genere provoca il cambiamento dello stato interno dell'artefatto ed eventualmente genera uno o più *observable event*, che possono

poi essere raccolti dai sensori degli agent body, ed infine essere percepiti dagli agenti attraverso esplicite operazioni di *sensing*.

L'interfaccia d'uso di un artefatto può cambiare in funzione dello *observable state*, esponendo differenti gruppi di operazioni a seconda dello stato interno. Il concetto di observable state viene utilizzato al fine di strutturare le modalità comportamentali dell'artefatto in gruppi di stati noti.

Un agente può invocare l'esecuzione di un'operazione su di un artefatto se e solo se l'operazione e correntemente compresa nell'interfaccia d'uso, in caso contrario l'azione avviata dall'agente si conclude con un fallimento.

Function description e operating instructions

Per garantire un utilizzio razionale degli artefatti da parte degli agenti, ognuno di essi è dotato di una *function description*, che esprime le funzionalità disponibili attraverso il suo utilizzo, e di *operating instructions*, che descrivono puntualmente *come* utilizzare l'artefatto per farlo assolvere alla sua funzione.

Queste descrizioni sono indispensabili per mettere gli agenti cognitivi presenti nel sistema di *(i)* determinare dinamicamente quale artefatto debba essere selezionato per il supporto alle loro attività, *(ii)* acquisire le capacità per comprendere in quale modo interagire in modo corretto.

3.1.3 Workspaces

Gli artefatti sono collocati all'interno di *workspaces*, che possono essere utilizzati per la definizione della topologia dell'ambiente di lavoro. Un workspace può essere definito come un insieme aperto di artefatti ed agenti che possono creare ed utilizzare: gli artefatti possono essere aggiunti o rimossi da un workspace e gli agenti possono dinamicamente entrarvi o uscirne.

In CArtAgO i workspace sono definiti da un nome logico ed un identificatore univoco. Una condizione necessaria per far sì che un agente possa utilizzare un artefatto è che quest'ultimo esista nel workspace in cui l'agente si trova. Dualmente, gli eventi generati da un artefatto potranno essere percepiti da un agente solo se questo si troverà all'interno dello stesso workspace.

È possibile ricorrere ad operatori di intersezione e nesting relativamente agli workspace, in modo da poter determinare topologie anche complesse. In particolare l'intersezione permette agli stessi artefatti ed agenti di appartenere a differenti workspace.

Artifact construction and disposal	`createArtifact(Name,Template,Config,{WsID}):ArID` `disposeArtifact(ArID)`

Artifact selection and use	`getArtifactID(Name,{WsID}):ArID`
	`execOp(ArID,OpName,{Args},{SensorID})`
	`sense({SensorID},{Pattern},{Timeout}):Perception`
	`focus(ArID,SensorID)`
	`unfocus(ArID,SensorID)`

Artifacts inspection	`getFD(ArID): FDDescr`
	`getOI(ArID): OIDescr`
	`getUID(ArID): UIDDescr`
	`getState(ArID): StateDescr`

Sensor management	`linkSensor(SensorType,SensorConfig): SensorID` `unlinkSensor({SensorID})`

Workspaces management	`getWsID(WsName):WsID`
	`createWS(WsName):WsID`
	`disposeWS(WsID)`
	`registerArtifact(ArID,WsID)`
	`deregisterArtifact(ArID,WsID)`
	`joinWS(WsID)`
	`exitWS(WsID)`

Tabella 3.1: Azioni a disposizione degli agenti per la gestione di artefatti e workspace

3.2 Core primitives

Dopo l'illustrazine delle principali caratteristiche e componenti di CArtAgO è utile analizzare il set di *core API* messe a disposizione dall'ambiente per permettere agli artefatti di essere utilizzati dagli agenti e per definirne struttura e comportamento.

3.2.1 Agent side

Dal punto di vista degli agenti sono previste una serie di primitive per il controllo degli agent body, che tendenzialmente ha ricadute pratiche in termini di operazioni compiute all'interno dell'ambiente.

Costruzione e dispose degli artefatti – Sono perviste due primitive di base per creare (`createArtifact`) ed effettuare la dispose (`disposeArtifact`) dinamiche degli artefatti. Per creare un artefatto è necessario fornire un nome logico, unitamente al `Template` che identifica il tipo di artefatto da creare, ed eventualmente il workspace in cui si intende crearlo. L'azione può fallire nel caso in cui il template non venga riconosciuto o l'istanziamento fallisca per qualche motivo.

Selezione ed utilizzo degli artefatti – Queste primitive sono fondamentale ai fini delle interazioni tra agenti ed artefatti, permettendo l'esecuzione di operazioni e l'osservazione dello stato degli artefatti e degli eventi da essi generati. Per eseguire un'operazione viene fornita la primitiva `exOp`, per la quale è necessario specificare l'indentificatore univoco dell'artefatto, il nome dell'operazione, i parametri ed eventualmente il sensore specifico nel quale raccogliere gli eventi osservabili eventualmente generati dall'artefatto in conseguenza dello svolgimento dell'operazione specificata. L'azione può fallire nel caso in cui l'artefatto non sia disponibile o perchè essa non può essere completata in quanto non attualmente presente nell'interfaccia d'uso dell'artefatto. L'identificatore di un artefatto esistente può essere ottenuta tramite la primitiva `getArtifactID`. specificandone nome logico ed eventualmente il workspace in cui esso è situato.

Dopo aver invocato un'operazione un agente può osservarne gli eventi conseguenti attraverso l'esecuzione di `sense` sul sensore specificato al momento dell'invocazione di `exOp`. In particolare, l'effetto di tale azione consiste nel rimuovere gli stimoli raccolti dal sensore trasferendoli all'agente sotto forma di percezioni, generando un fallimento nel caso in cui non ve ne siano all'interno del sensore. Sono previste diverse tipologie di sensore, inoltre, è possibile (opzionalmente) specificare un parametro temporale per indicare la durata dell'operazione di sensing, nel caso in cui non occorrano eventi nell'arco di tempo indicato si ha nuovamente un fallimento, l'intervallo di default è di durata nulla.

Al fine di permettere forme di sensing *data-driven* é possibile specificare un pattern, che svolge la funzione di filtro e seleziona quindi solo un sottoinsieme di eventi. Stimoli non considerati perchè scartati dal filtro non vanon comunque perduti, e possono essere percepiti in seguito attraverso nuove operazioni di sensing.

Sono infine previste primitive per l'osservazione continuativa di un artefatto: con l'operazione di `focus` un agente diviene osservatore persistente di un artefatto ed all'interno del sensore specificato vengono raccolti tutti gli eventi generati. Con la primitiva `unfocus` è possibile interrompere l'osservazione.

Ispezione degli artefatti – A supporto dell'utilizzo cognitivo degli artefatti è previsto un insieme di primitive per l'ottenimento di function description (`getFD`), operating instruction (`getOI`), user interface (`getUID`) e dello stato osservabile dinamico dell'artefatto (`getState`).

Gestione del sensore – Sono previste primitive per la connessione e disconnssione dinamica di sensori agli agent body: (`linkSensor`) connette un sensore del tipo specificato al body, restituendo l'identificatore necessario per le

Observable event generation	`genEvent({OpID},EventType,{EventContent})` `genEventInWsp(EventType,{EventContent})`
Operation management	`getOpID: OpID`
Observable state management	`setObservableState(ObsStateName)` `getObservableState:ObsStateName`

Tabella 3.2: Primitive di base per la programmazione degli artefatti

successive operazioni di sensing, con (`unlinkSensor`) è possibile disconnettere un sensore precedentemente aggiunto.

Gestione del workspace – Infine, sono previste primitive per la gestione della topologia logica dell'ambiente di lavoro. Esse permettono di entrare (`joinWS`) ed uscire (`exitWS`) da un workspace, di conoscerne l'identificatore (`getWsID`) a partire dal nome logico, di crearne di nuovi (`createWS`) o di rimuoverli completamente (`disposeWS`).

Siccome un artefatto può venire associato a più workspace, sono previste primitive per registrarlo ad un workspace (`registerArtifact`) e deregistrarlo (`deregisterArtifact`).

3.2.2 Artifact side

Dal punto di vista degli artefatti, CArtAgO mette a disposizione una serie di strumenti per la definizione di strutture e comportamenti.

Gli eventi osservabili possono essere generati sia in relazione alla specifica esecuzione di un'operazione, sia in modo ad essa svincolato. Per permettere questa distinzione ogni operazione eseguita sull'artefatto è identificata univocamente (`OpId`), attraverso la primitiva `getOpId` è poi possibile recuperare l'identificatore durante l'esecuzione.

Un evento può essere generato attraverso la primitiva `genEvent`, specificando l'identificatore dell'operazione a cui esso è relazionato. Se l'identificatore `OpId` non viene specificato l'evento si intende relativo all'operazione correntemente in fase di esecuzione. È possibile inoltre generare eventi non relazionati ad alcuna operazione attraverso la primitiva `getEventInWsp`, che genera un evento osservabile da tutti gli agenti che in precedenza avevano eseguito l'operazione di `focus` sull'artefatto.

Infine, sono previste due primitive per la gestione dello stato osservabile dell'artefatto, in particolare per impostare un nuovo stato (`setObservableState`) ed ottenere invece quello corrente (`getObservableState`).

Concludendo, si è detto che un evento può essere esplicitamente generato attraverso la primitiva `genEvent`, ma esistono anche una serie di eventi generati

implicitamente dal framework all'occorrenza di specifiche condizioni, come il completamento di un operazione, oppure il cambiamento dell'observable state di un artefatto.

Capitolo 4

Conclusioni

L'ampliamento della visione d'insieme nell'ambito della modellazione di sistemi multi-agente ha, con l'introduzione dell'astrazione di *artefatto*, sicuramente reso più agevole l'approccio progettuale a sistemi complessi. In particolare l'eliminazione dello stretto vincolo di proporzionalità esistente tra la complessità delle attività svolte e quella degli agenti coinvolti, rende possibile uno scaling più naturale e meno oneroso dal punto di vista implementativo.

La formalizzazione del meta-modello A&A ha messo a disposizione il riferimento ad uno schema infrastrutturale comune per la definizione dei sistemi e, soprattutto, delle interazioni al loro interno, permettendo di spostare l'attenzione sulle effettive dinamiche comportamentali più che sugli aspetti specificamente strutturali.

L'introduzione di *capacità cognitive* in agenti ed artefatti infine, ha garantito la possibilità di orientare la ricerca in direzione della realizzazione di entità, ed in generale di sistemi, che presentino caratteristiche tali da poter essere associate al concetto di *autonomia*.

In questo contesto, il framework CArtAgO rappresenta uno strumento essenziale e di immediata abilitazione pratica, grazie all'astrazione di alto livello offerta, ispirata dalle dinamiche osservate nella società umana. L'ortogonalità rispetto alla specifica architettura utilizzata per gli agenti[1] permette di accedere alle funzionalità del framework indipendentemente dalla legacy di provenienza e, soprattutto, di mettere in pratica forme concrete di cooperazione in cui agenti natura architetturale eterogenea interagiscano con i medesimi artefatti.

[1] attualmente (prima metà 2010, versione di riferimento: 1.4) sono supportate le architetture Jason, Jadex, simpA, 2APL e Java

Bibliografia

[1] A. Ricci, M. Viroli, A. Omicini, *"Programming MAS with Artifacts"* DEIS, Alma Mater Studiorum, Università di Bologna, 2005

[2] M. Viroli, A. Omicini, A. Ricci *"Engineering MAS environment with Artifacts"* DEIS, Alma Mater Studiorum, Università di Bologna, 2006

[3] A. Omicini, A. Ricci, M. Viroli *"Artifacts in the A&A meta-model for multi-agent systems"* DEIS, Alma Mater Studiorum, Università di Bologna, 2008

[4] A. Ricci, M. Viroli, A. Omicini *"*CArtAgO*: a framework for prototyping artifact-based environments in MAS "* DEIS, Alma Mater Studiorum, Università di Bologna, 2008

[5] A. Ricci (reference author) *"*CArtAgO *Annotated Manual"* aliCE team @ DEIS, Alma Mater Studiorum, Università di Bologna, 2006-2008

[6] CArtAgO project homepage - http://sourceforge.net/projects/cartago/

Elenco delle figure

Elenco delle tabelle