



**POLITECNICO DI MILANO**  
*DIPARTIMENTO DI ELETTRONICA E INFORMAZIONE*

---

# **Sistemi Eterogenei per la Collaborazione in Rete**

**Heterogeneous Systems  
For Collaboration Over The Internet**

**Thimoty Barbieri**

Relatore: Prof. Paolo Paolini

Dottorato di Ricerca in Ingegneria Informatica e Automatica 2000/2003



Politecnico di Milano  
*DIPARTIMENTO DI ELETTRONICA E INFORMAZIONE*  
Dottorato di Ricerca in Ingegneria Informatica e Automatica

---

# Sistemi Eterogenei per la Collaborazione in Rete

Tesi di Dottorato di Ricerca di:  
Dott. Ing. Thimoty Barbieri

**Relatore:** Prof. Paolo Paolini

**Coordinatore Dottorato:** Prof. Carlo Ghezzi

Il Dottorando

Il Tutore

Il Coordinatore

XV Ciclo



Politecnico di Milano  
*DIPARTIMENTO DI ELETTRONICA E INFORMAZIONE*  
Dottorato di Ricerca in Ingegneria Informatica e Automatica

---

# Heterogeneous Systems For Collaboration Over The Internet

PhD Thesis by:  
Thimoty Barbieri

**Advisor:** Prof. Paolo Paolini  
**Supervisor of the PhD Program:** Prof. Carlo Ghezzi

???????? ???? ??????  
???? ? ??? ??????  
???????, ?? ?????????  
????? ????? ???? ????????? ?????.

## Sintesi

*I sistemi collaborativi in rete supportano il lavoro di gruppo tra più utenti che non si trovano fisicamente nel medesimo luogo. L'oggetto di studio del nostro lavoro sono i sistemi collaborativi in rete che rappresentano le dinamiche di gruppo e lo spazio di lavoro, utilizzando una rappresentazione tridimensionale aggiornata in tempo reale. Una seconda caratteristica dei sistemi da noi studiati, è che consentono di accedere ad uno stesso spazio di lavoro utilizzando canali diversi, cioè sistemi di accesso diversi in termini di tecnologia usata e di rappresentazione dello spazio di lavoro. I problemi che emergono nello studio e nella realizzazione di questa categoria di sistemi sono sia concettuali che implementativi.*

*Dal punto di vista concettuale, proponiamo alcuni modelli che devono fornire una guida nella analisi e nella progettazione di un sistema collaborativo tridimensionale multicanale. Questi modelli illustrano come avviene la collaborazione tra utenti e secondo quali canoni può essere normata e diretta (modello delle metafore cooperative); come organizzare lo spazio tridimensionale per renderlo strutturato e navigabile (modello delle metafore visive); come fare in modo che lo spazio tridimensionale possa essere utilizzato come uno spazio di accesso tridimensionale verso contenuti bidimensionali (foto, testi, audio, video) raccolti in un sito web tradizionale (progettazione di strutture di accesso descritte con un opportuno DTD); e per finire quali criteri seguire per valutare le caratteristiche di un canale e progettarlo per interagire con gli altri canali (modello della pila della multicanalità).*

*Accanto ai modelli concettuali, si porta l'attenzione sui maggiori problemi implementativi posti da questa categoria di sistemi: la difficoltà nel rendere in tempo reale una visualizzazione tridimensionale grafica dello spazio di lavoro, come assistere gli utenti nell'uso della collaborazione all'interno degli ambienti, e come effettivamente realizzare canali alternativi che sfruttino la rete radiomobile in congiunzione con un terminale cellulare o un PDA. Un altro problema riguarda la necessità di creare situazioni che scatenino collaborazione, laddove non ci sia: l'innescò di attività collaborativa è necessario in quegli ambienti che si prefiggono di richiamare utenti ad una visita, piuttosto che offrire un mezzo specializzato per condurre il proprio lavoro. A questo scopo si è analizzato l'effetto derivante dall'uso di agenti automatici programmati all'interno degli ambienti virtuali collaborativi. Per quanto riguarda la rappresentazione grafica, viene proposto un prototipo di motore grafico strutturato ad oggetti mediante le API Java3D. I concetti cardine utilizzati da questo motore sono la gestione dinamica del Grafo di Scena rappresentante lo spazio collaborativo tridimensionale, e la gestione dinamica di Comportamenti, classi di codice mobile dinamico, che possono essere collegate alla scena a runtime per descrivere i comportamenti degli elementi del mondo. Viene inoltre proposto uno schema di framework generale, che organizza tutti gli aspetti caratteristici della categoria di sistema in esame. Per ciascun aspetto del framework generale proposto, si è realizzata una soluzione prototipale che ha consentito di derivare risultati e osservazioni. I modelli concettuali derivati dall'esperienza prototipale consentono di comprendere meglio quali problematiche dovranno affrontare questi sistemi nel futuro, quando superate alcune barriere tecniche, la loro presenza in contesti multicanali diverrà sempre più concreta.*

## Indice

<b>1</b>	<b>OBIETTIVI .....</b>	<b>1</b>
<b>2</b>	<b>I SISTEMI COLLABORATIVI .....</b>	<b>2</b>
2.1	LA PERCEZIONE DELLO SPAZIO CONDIVISO .....	2
2.2	LA COMUNICAZIONE NELLO SPAZIO CONDIVISO .....	5
2.3	APPLICAZIONI ED APPROCCI NELLA TECNOLOGIA ATTUALE .....	7
2.4	COOPERAZIONE MEDIANTE MEZZI VISUALI .....	10
2.5	METAFORE VISIVE .....	13
2.6	ARCHITETTURA GENERALE DI UN SISTEMA COLLABORATIVO MULTICANALE .....	19
2.6.1	<i>WebTalk-I</i> .....	20
2.6.2	<i>WebTalk-II</i> .....	21
<b>3</b>	<b>LE METAFORE COOPERATIVE .....</b>	<b>27</b>
3.1	IL CONCETTO DI METAFORA COOPERATIVA .....	27
3.2	IL MODELLO BASE .....	30
3.2.1	<i>Distribuzione dello stato condiviso</i> .....	31
3.2.2	<i>Meccanismi di Raggruppamento</i> .....	32
3.2.3	<i>Flusso Informativo</i> .....	33
3.2.4	<i>Visualizzazione</i> .....	34
3.2.5	<i>Movimento</i> .....	35
3.2.6	<i>Autorappresentazione</i> .....	35
3.2.7	<i>Comunicazione Testuale</i> .....	36
3.2.8	<i>Esempio di Metafora Cooperativa</i> .....	37
3.3	IL MODELLO ESTESO .....	38
3.3.1	<i>Definizioni</i> .....	38
3.3.2	<i>Esempio applicativo</i> .....	45
<b>4</b>	<b>LA GESTIONE DEI CONTENUTI ED IL WEB COLLABORATIVO .....</b>	<b>50</b>
4.1	PREMESSA .....	50
4.2	LA MODELLAZIONE LOGICA E LE STRUTTURE DI ACCESSO .....	51
4.3	LA NAVIGAZIONE 3D SUI CONTENUTI .....	52
4.4	CONFIGURATORE 3 D .....	54
4.5	FORMATI DI PUBBLICAZIONE E CONFIGURAZIONE .....	63
4.6	CONCLUSIONI .....	69

<b>5</b>	<b>LA RAPPRESENTAZIONE GRAFICA TRIDIMENSIONALE</b>	<b>70</b>
5.1	PREMESSA .....	70
5.2	IL KERNEL GRAFICO JAVA 3D .....	70
5.2.1	<i>Caratteristiche di Java 3D</i> .....	70
5.2.2	<i>Scene Builder: creazione dinamica degli universi virtuali</i> .....	75
5.2.3	<i>Conclusioni sull'uso di Java 3D</i> .....	89
5.3	TECNICHE DI RENDERING PRESTAZIONALE .....	89
5.4	CONCLUSIONI.....	94
<b>6</b>	<b>LA COLLABORAZIONE ASSISTITA .....</b>	<b>95</b>
6.1	NECESSITÀ DI COLLABORAZIONE ASSISTITA.....	95
6.1.1	<i>Assistenza di base</i> .....	98
6.1.2	<i>Guida Applicativa</i> .....	100
6.1.3	<i>Consulenza</i> .....	100
6.1.4	<i>Popolazione e Dimostrazione</i> .....	101
6.2	IMPLEMENTAZIONE DI UN SISTEMA ASSISTIVO ALLA COLLABORAZIONE.....	102
6.2.1	<i>Funzionalità di Aiuto</i> .....	102
6.2.2	<i>Funzionalità di guida automatica</i> .....	105
6.2.3	<i>Funzionalità di popolamento</i> .....	109
6.2.4	<i>Architettura generale dei moduli agente</i> .....	112
6.3	CONCLUSIONI.....	115
<b>7</b>	<b>LA MULTICANALITÀ L'USO DI DEVICE MOBILI.....</b>	<b>116</b>
7.1	FONDAMENTALI CARATTERISTICHE DELLA MULTICANALITÀ IN CONTESTO MOBILE .....	116
7.1.1	<i>La pila della multicanalità</i> .....	116
7.1.2	<i>Presentazione</i> .....	118
7.1.3	<i>Navigazione</i> .....	119
7.1.4	<i>Dominio Funzionale</i> .....	121
7.1.5	<i>Collaborazione Multicanale</i> .....	121
7.2	IMPLEMENTAZIONE DEL CANALE WAP .....	123
7.2.1	<i>Esempi di collaborazione sul WAP</i> .....	123
7.2.2	<i>Analisi del canale WAP</i> .....	125
7.2.3	<i>Esempio applicativo</i> .....	135
7.2.4	<i>Considerazioni sui Risultati Ottenuti</i> .....	141
7.3	UTILIZZO DI DISPOSITIVI PALMARI .....	142

7.4	IMPLEMENTAZIONE DEL CANALE PER PALMARE .....	151
7.4.1	<i>Architettura fisica</i> .....	151
7.4.2	<i>Strato di Presentazione</i> .....	153
7.4.3	<i>Strato di Navigazione</i> .....	158
7.4.4	<i>Strato funzionale</i> .....	166
7.4.5	<i>Strato collaborativo</i> .....	167
7.4.6	<i>Esempio applicativo</i> .....	168
<b>8</b>	<b>CONCLUSIONI E SVILUPPI FUTURI</b> .....	<b>174</b>
<b>9</b>	<b>RINGRAZIAMENTI</b> .....	<b>176</b>
<b>10</b>	<b>BIBLIOGRAFIA</b> .....	<b>177</b>

# 1 Obiettivi

Lo scopo del presente lavoro di Ricerca è di esplorare la natura della collaborazione remota tra più persone intente ad una attività utilizzando strumenti di lavoro interconnessi attraverso reti Internet o intranet.

I nostri obiettivi sono la definizione di modelli concettuali che consentano di descrivere i meccanismi con cui questa collaborazione avviene, e che spieghino quali siano le migliori modalità per supportarla, a prescindere dalla tecnologia utilizzata.

In particolare, uno dei primi intenti è di esplorare come esprimere la collaborazione tra più utenti remoti utilizzando una rappresentazione tridimensionale del contesto di collaborazione. Il secondo intento riguarda invece la definizione di un modello di analisi di applicazioni per la collaborazione in un contesto multicanale, a cui cioè sia possibile accedere mediante tecnologie e strumenti diversi.

L'elaborazione di modelli generali con cui trattare il problema della collaborazione tra utenti remoti deve passare attraverso la realizzazione sperimentale di prototipi che consentano di realizzare effettivamente sessioni di collaborazione remota. L'analisi dei risultati ottenuti dall'utilizzo di questi prototipi, e l'esperienza accumulata nella loro progettazione e realizzazione, deve servire per fornire ulteriori elementi per aumentare la comprensione del problema, ed arrivare a definire un modello generale che descriva la struttura concettuale, il funzionamento, e i principali aspetti implementativi di un sistema collaborativo di rete multicanale, con possibilità di rappresentazione tridimensionale dell'area di lavoro.

## 2 I sistemi collaborativi

### 2.1 La Percezione dello Spazio Condiviso

Le tecnologie informatiche per gruppi di lavoro collaborativi (Computer Supported Collaboration Workgroups, CSCW) sono in costante sviluppo grazie ai continui progressi tecnologici nell'ambito delle telecomunicazioni, dell'informatica, e della conoscenza dello strumento informatico da parte di ogni singolo utente. In un contesto in cui il rapido sviluppo di Internet ha consentito un miglioramento delle comunicazioni a livello globale, sempre maggiore è l'esigenza di comunicare, lavorare ed interagire con altre persone anche quando non sia fisicamente possibile condividere lo spazio di lavoro sedendosi attorno ad un tavolo.

Tutte le tecnologie CSCW si prefiggono dunque questo obiettivo: fornire una tecnologia di supporto, ma anche e soprattutto una metodologia di lavoro, per consentire a più persone di portare a termine insieme, collaborando, un'attività, senza dover essere fisicamente nel medesimo posto.

Le applicazioni delle tecnologie CSCW sono molteplici e tutte di estremo interesse: si va dall'e-conferencing, piattaforme di supporto a riunioni e discussioni, ai giochi cooperativi, ad applicazioni quali l'e-learning, dove la piattaforma di collaborazione supporta l'erogazione di contenuti didattici da parte di uno o più docenti verso una moltitudine di studenti remoti.

Altre applicazioni riguardano l'e-shopping, dove la piattaforma di collaborazione serve a far incontrare l'acquirente con consulenti all'acquisto, laddove il prodotto venduto richieda un certo grado di competenza tecnica. Si possono pensare poi ambiti in cui si sfrutta la possibilità di generare turismo o educazione virtuale, mediante la costruzione di musei virtuali, a cui tutti i visitatori del mondo possono accedere condividendo le proprie esperienze con gli altri o con le guide del museo stesso.

I CSCW sono realizzabili tecnicamente in numerosissimi modi, e consentono di offrire modalità e paradigmi diversi per incoraggiare e rappresentare la collaborazione in atto tra i partecipanti all'attività gestita dalla piattaforma di collaborazione. Tuttavia, tutti i sistemi di collaborazione sono realizzati cercando di realizzare i medesimi principi fondanti, e di sfruttare le medesime caratteristiche che contraddistinguono il modo con cui i partecipanti comunicano e lavorano nella vita reale. L'intento principale dei framework CSCW è quello di *rappresentare* ed offrire meccanismi che si ispirano al modo con cui le persone collaborano nella realtà, eventualmente partendo da questi meccanismi per offrire modalità di interazione nuove, non possibili nel reale, ma che sfruttano la tecnologia e le caratteristiche del framework, fornendo quindi un valore aggiunto dell'uso del CSCW rispetto all'interazione e alla collaborazione fisica.

L'aspetto fondamentale di un lavoro in una dinamica di gruppo è la *Workspace Awareness*<sup>1</sup>, vale a dire la percezione istantanea di stare interagendo con una o più persone all'interno di uno spazio di lavoro condiviso. Questa sensazione si compone di alcune informazioni di base, quale la nozione di dove sono le persone con cui si sta collaborando all'interno dello spazio (ad es. dall'altro capo del tavolo da riunione), di che cosa stanno facendo, e di quello che presumibilmente faranno nell'immediato futuro. Sono queste informazioni che consentono di coordinare le azioni tra i partecipanti, discutendo od eseguendo insieme un'attività, aiutandosi l'un l'altro.

E' interessante inoltre osservare come una delle prime conseguenze della Percezione dello Spazio di Lavoro sia la capacità, da parte di ciascuno dei partecipanti, di prevedere in un breve spazio temporale quali saranno le azioni intraprese dagli altri, e come sia questa, da un lato, una delle chiavi dell'effettivo svolgersi della collaborazione, e dall'altro il meccanismo attraverso il quale il soggetto percepisce di stare effettivamente collaborando. In altre parole, è la netta sensazione di poter prevedere cosa succederà nell'immediato a fornire la certezza di stare collaborando in modo efficace nello svolgersi dell'attività.

La Percezione dello Spazio Condiviso si articola tipicamente su degli aspetti cognitivi incentrati sul "chi", sul "dove", sul "cosa", e sul "quando".

- L'aspetto soggettivo ("chi") consente di enumerare i partecipanti e determinare chi essi siano e quali siano le loro caratteristiche. Esso fornisce informazioni riguardo alla presenza e all'identità di ciascun partecipante.
- L'aspetto spaziale ("dove") consente ai partecipanti di determinare la propria posizione nello spazio condiviso rispetto ai propri partecipanti. In un contesto reale, questo può significare l'essere consci di sedere ad un tavolo, alla destra del Vicepresidente, con l'influenza che questo può avere sul modo con cui condurre le attività durante la sessione di lavoro.
- La sfera spaziale riguarda anche aspetti quali lo sguardo (dove stanno guardando i partecipanti?), la vista (fino a dove possono vedere), o l'area di azione (fino a dove possono andare/interagire).
- L'aspetto oggettivo ("cosa") riguarda la percezione esatta di quali prodotti, concetti od oggetti sono stati realizzati, proposti o manipolati da quali persone, sia nell'immediato passato, che nel momento presente, che nel successivo futuro. Questo corrisponde a sapere che il foglio dell'ordine del giorno è stato distribuito a tutti, ma che la cartellina rossa con documenti riservati è stata distribuita solamente alle tre persone che siedono alla testa del tavolo.
- L'aspetto temporale ("quando") riguarda la percezione di come la collaborazione si è svolta, si sta svolgendo o si svolgerà lungo un arco temporale. In altre parole, questo corrisponde a rendersi conto che uno dei partecipanti, dopo aver sottoposto all'attenzione di tutti alcuni lucidi, non è più intervenuto nella conversazione da almeno mezz'ora, fatto da cui si può

---

<sup>1</sup> Il concetto di *Percezione dello Spazio Condiviso* è tratto da Gutwin C., Greenberg S. – A framework of Awareness for Small Groups in Shared Workspace Groupware. *Technical Report 99-1, Dept. of Computer Science, University of Saskatchewan, Canada, 1999* e successivamente rielaborato sulla scorta delle nostre personali esperienze nel campo.

dedurre un suo disinteresse all'andamento dei lavori, una sua disapprovazione, o imbarazzo.

Si vede dunque come la Percezione dello Spazio Condiviso, articolata sui quattro aspetti "chi", "dove", "cosa", "quando", pur non fornendo informazione direttamente oggetto del lavoro e dell'attività del gruppo, è indispensabile per costruire tutta una rete di informazioni collaterali che consentono a ciascun partecipante di meglio comprendere la dinamica dell'attività, e di scegliere un corso di comportamento nell'esecuzione dell'attività stessa, che non sarebbe stato considerato in un contesto diverso o senza le informazioni derivate dalla percezione della collaborazione.

Per poter dunque fornire ai partecipanti di un framework collaborativo la percezione di collaborazione nel modo più netto possibile, occorre strutturare la piattaforma attorno a tre aspetti principali:

- Quali sono le informazioni rilevanti di cui i partecipanti devono tenere traccia per poter svolgere l'attività in modo collaborativo
- Quali sono gli elementi che trasmettono Percezione di Spazio Condiviso ai partecipanti nel contesto
- Come i partecipanti utilizzano la Percezione dello Spazio Condiviso per tradurla in collaborazione

Questi tre aspetti dovrebbero essere scelti come linee guida nella realizzazione della piattaforma di collaborazione, rispettivamente per selezionare quali sono le informazioni che il sistema deve selezionare e distribuire ai partecipanti, quali quelle che, pur non essendo direttamente parte dell'obiettivo dell'attività da svolgere, servono a supportare la nozione di collaborazione, e, per finire, quali sono i meccanismi e le funzioni che la piattaforma mette a disposizione dei partecipanti per reagire nel contesto collaborativo.

Per fissare le idee con un esempio, in un framework collaborativo che consente di parlare ed eseguire disegni su un foglio di carta comune, il sistema dovrà distribuire la conversazione testuale (chat), e rappresentare in tempo reale i disegni su un'opportuna area bianca dello schermo. Se queste sono le effettive informazioni scambiate nell'ambito dell'attività, esse non sono sufficienti per dare una percezione di spazio condiviso. Occorrerà visualizzare, ad esempio, delle "matite" digitali su cui è posto il nome di ciascuno dei partecipanti, assegnare un diverso colore a ciascuno dei segni tracciati dai partecipanti, indicare il tempo di inattività (idle time) di ciascun partecipante nella discussione. Questi elementi consentono di capire in modo più preciso lo stato di ciascuno dei partecipanti esattamente secondo le modalità tipiche della Percezione della Condivisione.

Infatti, marcare la posizione della matita digitale con il nome dell'utente, rivela il "dove" ciascun utente si situa all'interno dello spazio condiviso, e dunque fornisce l'informazione spaziale necessaria alla corretta percezione di spazio collaborativo; l'uso di colori diversi nel tracciamento dei disegni consente di stabilire il "cosa" ciascun partecipante ha fatto negli istanti precedenti e sta facendo nell'istante corrente; per

finire, l'indicazione del tempo di inattività consente di stabilire "quando" un partecipante sta interagendo, o quanto frequentemente.

Per terminare l'esempio, questo ipotetico framework collaborativo dovrà fornire:

- una linea in cui digitare testo
- uno o più strumenti per tracciare all'interno del foglio di lavoro
- una lista di tutti i partecipanti aggiornata in tempo reale, da cui è possibile ricavare uno storico degli interventi di quel partecipante
- la possibilità di dichiarare il proprio stato (in attesa, in ascolto, non presente)
- ...

Questi strumenti esemplificano il terzo aspetto, vale a dire quali sono le funzioni messe a disposizione per "reagire", chiudere l'anello della percezione della collaborazione, alimentando la propria percezione e quella degli altri, mediante l'inserimento di nuovi stimoli di attività diretti agli altri partecipanti.

La percezione della collaborazione, dunque, si risolve nella conoscenza dello stato istantaneo di un ambiente definito, in termini di tempo e spazio. Questo ambiente evolve in modo continuo nel tempo per effetto dell'evolvere dell'attività portata avanti dai suoi partecipanti, e questo implica che questi cambiamenti devono essere continuamente rilevati e portati a conoscenza, nel medesimo modo, a tutti i partecipanti. Questo aspetto, il *mantenimento dello stato condiviso*, è la prima preoccupazione nell'implementazione di un framework collaborativo, e può essere realizzato con svariate tecniche.

Lo stato condiviso rappresenta la base su cui i partecipanti effettuano l'interazione e l'attività, ed è la conoscenza istantanea dello stato condiviso e l'interazione che da esso si genera, a creare la Percezione di Spazio Condiviso, necessaria alla collaborazione. Si crea dunque un "anello percettivo-interattivo", in cui la percezione dello stato condiviso genera interazione da parte del partecipante, e questa interazione alimenta l'evoluzione dello stato condiviso, aumentando la percezione di condivisione da parte di tutti gli altri partecipanti.

Un ulteriore elemento da considerare in questo contesto è lo scopo finale dell'interazione. L'interazione, infatti, riguarda sempre una determinata attività da compiere: la collaborazione non è mai il fine, ma un mezzo. Il contesto collaborativo ed i meccanismi di collaborazione offerti sono progettati in modo da far tendere il gruppo di lavoro verso il proprio obiettivo.

## **2.2 La comunicazione nello spazio condiviso**

Lo scopo principale per cui una attività si svolge in un contesto collaborativo, sia nell'ambito reale che nell'ambito di piattaforme basate su reti di elaboratori, è quello di fornire ai partecipanti dei meccanismi avanzati di comunicazione. La collaborazione, infatti, funge da substrato su cui è possibile realizzare elementi di comunicazione avanzata, non possibili senza la condivisione di uno spazio di lavoro.

La progettazione di un framework di lavoro condiviso dovrà, dunque, tenere presente che lo scopo della creazione di una percezione di spazio condiviso è l'ottenimento di

queste tecniche di comunicazione avanzate, e tentare di far sì che i partecipanti possano sfruttarle.

Il primo aspetto comunicativo che si instaura all'interno di un meccanismo collaborativo, oltre a quello verbale/testuale ottenibile in normali comunicazioni remote, è la *comunicazione consequenziale*<sup>2</sup>: vale a dire la trasmissione di informazioni come conseguenza delle attività all'interno dell'ambiente. Si tratta di un mix di messaggi inviati mediante gestualità, tono della conversazione, movimenti, ecc., che hanno però un carattere non intenzionale od esplicito. Altre forme sono la *comunicazione laterale*, in cui la raccolta di informazioni avviene percependo una comunicazione che non è diretta ad un particolare destinatario, ad esempio sentita in corridoio oppure mentre una persona mormora a se stessa.

Se da una parte dunque il tessuto collaborativo può fornire maggiori informazioni, dall'altra semplifica la veicolazione dell'informazione stessa, mediante diverse altre tecniche:

- *Riferimento deittico*: consiste nell'usare all'interno della propria comunicazione riferimenti diretti all'ambiente condiviso (vai *là adesso* a prendere *quello* e mettilo *qui* accanto a *questo*). In questo modo la conversazione diventa molto più rapida e meno complessa, ma richiede che entrambi gli interlocutori abbiano modo di risolvere (spazialmente e temporalmente) il riferimento deittico.
- *Dimostrazione*: si tratta di usare la propria gestualità per rappresentare, indicare o spiegare un determinato concetto, ad esempio la serie standard di gesti adottati dagli steward come spiegazione accompagnatoria delle misure di sicurezza a bordo di aeromobili.
- *Manifestazione di un'azione*: consiste nell'espletare un'azione per portare un significato, ad esempio appoggiare un prodotto vicino al registratore di cassa per esprimere la volontà di acquistarlo.
- *Osservazione visiva*: consiste nell'usare la percezione visiva dell'ambiente per dirigere la propria comunicazione, ad esempio in conversazioni del tipo: "un po' più avanti, a destra, ancora un poco, lentamente, ecco, fermati".

L'innesco di queste modalità di comunicazione è, dunque, segno inequivocabile della presenza di percezione dello spazio condiviso da parte degli utenti. Un sistema di cooperazione che riuscisse a realizzare una collaborazione talmente sofisticata da convincere i propri partecipanti ad usare questi meccanismi di comunicazione con naturalezza, avrebbe raggiunto un grande risultato: supportare un lavoro di gruppo, con le stesse modalità di comunicazione di un gruppo di lavoro reale, ma avendo i partecipanti a grande distanza tra di loro.

---

<sup>2</sup> Segal L., *Designing Team Workstations: The Choreography of Teamwork*, in *Local Applications of the Ecological Approach to Human-Machine Systems*, P. Hancock, J. Flach, J. Caird, and K. Vicente ed., 392-415, Lawrence Erlbaum, Hillsdale, NJ, 1995.

Fin qui per quanto riguarda la collaborazione tra gruppi in numero piuttosto contenuto. In questi contesti le persone hanno ciascuna diretta percezione dell'altra, e calibrano la propria collaborazione sulle azioni di ciascuno degli altri utenti.

In situazioni di popolazione di massa, dove cioè il numero di utenti contemporanei è molto elevato, gli utenti passano attraverso una fase di esplorazione solitaria, dove solitamente rimangono ad osservare o ad ascoltare l'andamento della collaborazione. Questa fase consente di raccogliere rapidamente informazioni circa la natura dell'ambiente collaborativo e le tematiche in esse trattate, ma si tratta di una fase transitoria che porta pressoché invariabilmente alla formazione di sottogruppi più piccoli di utenti, in cui avviene una effettiva collaborazione in termini di mutua comunicazione finalizzata ad un obiettivo, e di un'eventuale interazione su contenuti o su oggetti.

In altre parole, sembra esserci una soglia limite di popolazione a cui i gruppi di collaborazione tendono. Superata questa soglia limite di utenti, si verifica invariabilmente e spontaneamente un fenomeno di disgregazione che crea sottogruppi di collaborazione. Si tratta di un andamento fisiologico, sostanzialmente dovuto a un limite cognitivo nella nostra capacità di attenzione diversificata: oltre un certo limite di interlocutori od azioni, non risulta possibile o efficace portare a termine una conversazione o una operazione. In definitiva, la teoria della collaborazione a piccoli gruppi rimane valida anche nella analisi e nel supporto di situazioni collaborative di massa.

### ***2.3 Applicazioni ed approcci nella tecnologia attuale***

Le attuali tecnologie disponibili all'interno delle reti di calcolatori (LAN e Internet) sono in grado di realizzare, con livelli differenti di complessità, un substrato metodologico e tecnologico che consente diversi gradi di collaborazione.

Analizzeremo ora alcune di queste tecnologie, con lo scopo di comprendere come esse riescano a fornire percezione di condivisione ed incoraggiare la comunicazione collaborativa – nel senso descritto precedentemente – e dove, invece, risultino carenti.

#### *Forum e chat (webboard, irc, icq...)*

Un semplice esempio di collaborazione tra più utenti viene realizzato attraverso l'utilizzo di forum<sup>3</sup> e di chat, oppure mediante instant messengers<sup>4</sup>.

Nell'esempio del forum, ciascun partecipante è identificato da un nome utente, corredato eventualmente da ulteriori informazioni personali descrittive, che consentono ad altri partecipanti una migliore conoscenza dell'utente stesso. Solitamente i forum sono organizzati in conferenze, gruppi di messaggi tematici eventualmente regolamentati da un moderatore, che si occupa di controllare che le tematiche di discussione siano pertinenti all'argomento della conferenza. Una discussione può essere organizzata in *thread* di discussione, per cui in ogni momento è possibile ripercorrere la catena di risposte e riferimenti dei messaggi che vengono inviati all'interno del forum.

---

<sup>3</sup> WebBoard: <http://webboard.oreilly.com/>

<sup>4</sup> ICQ: <http://www.icq.com/>

I forum sono una forma di cooperazione priva di una dimensione temporale immediata (i messaggi possono essere inoltrati in risposta anche a molte ore di distanza uno dall'altro), ma tuttavia consente di fornire ai partecipanti una modesta impressione di collaborazione.

La possibilità di identificare più partecipanti risponde alla dimensione soggettiva della collaborazione, l'organizzazione in thread dei contenuti dà una dimensione temporale dell'oggetto della collaborazione.

E' inoltre importante da notare la figura del moderatore, come arbitro della collaborazione che si instaura tra i partecipanti alla conferenza. Un modello del tutto simile è quello della chatroom (in stile IRC), dove la dimensione temporale della collaborazione risulta più rapida, poiché i messaggi sono brevi e scambiati in tempi rapidi (praticamente in tempo reale). Tuttavia il paradigma collaborativo rimane simile: si tratta di un sistema di comunicazione a canali, regolato da un moderatore, dove l'oggetto dell'attività è svolgere una discussione.

Un modello diverso di collaborazione è l'instant messaging: in questo paradigma la collaborazione avviene uno-a-uno, senza il concetto di canale di discussione o conferenza. Non esiste un concetto di moderatore, tuttavia questo tipo di collaborazione risulta interessante, poiché essendo incentrata sul particolare aspetto della gestione di brevi messaggi in tempo reale, risulta molto più articolata la gestione dello stato di attenzione ed interazione di ciascun utente con il sistema. Per questo motivo, non è infrequente trovare nella maggior parte dei software di instant messaging funzionalità che consentono di esprimere il proprio grado di collaborazione: "Not Available", "Away", "Do Not Disturb", a significare che il grado di interazione al momento possibile con questo utente è nullo o minimo. "Available", "Free For Chat", ecc., sono altri segnali che invece manifestano una piena apertura alla collaborazione o comunicazione.

In queste due modalità di collaborazione, il pregio sta nella semplicità e nell'ormai relativamente diffusa coscienza di esse tra gli utilizzatori della rete. Tuttavia, non esiste una vera e propria gestione di alcuni aspetti importanti dell'awareness, quale quello spaziale (non è possibile determinare *dove* si trovi un partecipante, nè esattamente dire che *cosa* faccia in un determinato momento).

### *Whiteboarding e teleconferencing (NetMeeting, PalTalk...)*

Sono una classe di applicativi collaborativi che propongono un lavoro di gruppo su contenuti grafici bidimensionali, solitamente un'area di disegno bianca su cui è possibile scrivere o disegnare schemi, oppure una condivisione del desktop di un computer o di una applicazione, per dimostrarne l'utilizzo. Spesso aggiungono alla funzionalità di chat ad una linea, corredata dal nome di ciascun partecipante, anche funzionalità di comunicazione audio e video.

In questo tipo di applicazioni, si utilizzano spesso più cursori di colore diverso, o recante ciascuno il nome di un diverso partecipante, per indicare la mutua posizione all'interno dello spazio bidimensionale condiviso. Esse sono dunque interessanti dal

punto di vista della collaborazione, poiché aggiungono una – seppur limitata – informazione spaziale, che aumenta la sensazione di condivisione.

E' da notare, poi, come l'uso del canale vocale risulti critico all'interno di un framework CSCW collaborativo. Il dialogo, infatti, viene quasi sempre utilizzato quando la comunicazione avviene tra due interlocutori, ma non in gruppo, dove il suo coordinamento creerebbe confusione. Un'eccezione interessante e' costituita dal sistema di comunicazione vocale PalTalk<sup>5</sup>, dove è possibile eseguire comunicazioni vocali anche in un largo gruppo, usando il tasto "Alzata di mano", e prenotando il proprio turno a parlare, proprio come in una situazione reale.

### *Piattaforme di E-learning (LearningSpace...)*

Una delle applicazioni collaborative di interesse sempre crescente riguarda l'e-learning, vale a dire la possibilità di realizzare didattica con allievi a distanza. Una "piattaforma" di e-learning è una metodologia che utilizza varie tecniche per cercare di fornire ausili didattici agli allievi, ricreando l'illusione di una "classe virtuale". Si tratta per lo più di soluzioni basate su pagine web dinamiche (ASP), che raccolgono in un modo strutturato le unità didattiche componenti i corsi. Ciascuno studente, dotato di un opportuno account e relativa password, si collega alla piattaforma e studia il materiale fornito attraverso il sito web. Durante la sua attività di studio, possono venire registrati parametri come l'ora e la data in cui ciascuna unità didattica viene cominciata, o se del determinato materiale di studio è stato scaricato, oppure ancora un compito assegnato è stato inviato alla piattaforma per essere valutato dai docenti. Le piattaforme offrono poi quiz di valutazione, i cui risultati sono resi noti allo studente, al docente, ed eventualmente ad altri studenti.

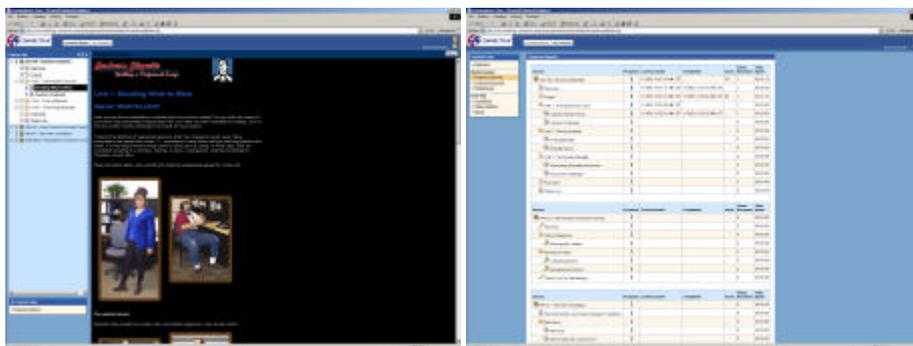
In quasi tutte le piattaforme sono forniti meccanismi di chat e di forum che consentono ad uno studente di dialogare con altri studenti, oppure con un tutor (una figura intermedia che serve a limitare il traffico di richieste diretto al docente), oppure ancora direttamente con il docente.

L'interesse all'aspetto collaborativo in questo tipo di piattaforme è molto elevato, poiché uno degli obiettivi è quello di cercare di evitare un "effetto isolamento" dato dalla formazione a distanza. Spesso infatti si programmano sessioni interattive o broadcast live di spiegazioni del docente, atti a cercare di dare una sensazione di coralità nella preparazione (lo studente percepisce che ci sono altri studenti, e che c'è effettivamente un docente che monitora la sua preparazione).

Un aspetto interessante nel costruire questa percezione di condivisione è la pubblicazione degli andamenti del corso da parte della classe di studenti, presente in alcune piattaforme. Sebbene si possa nutrire qualche riserva riguardo a questa scelta, data la natura delicata dell'informazione, tuttavia si può comprendere come queste informazioni, insieme al confronto tra studenti che nasce dalle discussioni nei forum e nelle chat, sia proprio rivolta a creare uno "spirito di corpo" tra gli studenti, necessario ad affrontare lo studio delle varie unità didattiche.

---

<sup>5</sup> [www.paltalk.com](http://www.paltalk.com)



**Fig 2-1: una piattaforma di Elearning**

### *Social Communities (3D)*

Un nuovo sistema di supportare la collaborazione è di fornire un canale di conversazione via chat, corredato da una rappresentazione tridimensionale. In essa viene visualizzato uno scenario fantastico o reale (spesso denominato “mondo”), in cui ciascun utente può muoversi, rappresentato da una “pedina” tridimensionale denominata “avatar” (il nome tecnico dell’anima reincarnata secondo la religione Buddhista). Gli avatar navigano nello spazio tridimensionale e possono compiere movimenti e azioni (danzare, toccare oggetti, saltare da una posizione all’altra del mondo). Lo scopo principale di queste applicazioni, almeno di quelle che vanno per la maggiore in Rete, sembra essere di interazione e comunicazione. Alcuni esperimenti riguardano il racconto di fiabe interattive, il turismo virtuale, l’archeologia virtuale. Tuttavia l’effettiva maturità di queste applicazioni, denominate collettivamente CVE (Collaborative Virtual Environment) o Net-VE, sembra ancora non raggiunta, e sembra che il 3D non riesca ad aggiungere effettivamente molto alle possibilità di intercomunicare tra gli utenti, oltre al creare un contesto comune (l’ambiente) con cui iniziare una conversazione, e a fornire delle vivide rappresentazioni grafiche che possono attirare l’attenzione dell’utenza.

Questi tipi di sistema rappresentano una interessante soluzione nell’ambito dei sistemi di collaborazione, ed il punto di partenza attorno a cui verte la nostra ricerca. L’utilizzo delle rappresentazioni 3D, infatti, consente di rappresentare secondo modalità nuove i meccanismi di collaborazione in rete già esistente, e di realizzare nuove, potenziate forme di collaborazione, il cui scopo è quello di aumentare ulteriormente la consapevolezza di essere all’interno di uno spazio condiviso.

## **2.4 Cooperazione mediante mezzi visuali**

Nella maggior parte delle applicazioni di collaborazione brevemente elencate in precedenza, lo spazio in cui gli utenti esercitano la propria facoltà di collaborare tra loro

è rappresentato utilizzando una impostazione grafica prevalentemente bidimensionale. Questo significa che si utilizzano i simboli, le icone e le azioni ormai comunemente accettate nella comunità degli utenti informatici. Un rettangolo grigio con un effetto ombreggiato rappresenta un pulsante e richiama un'operazione di click; un testo sottolineato rappresenta un link, un rettangolo con un elenco e una barra di scorrimento a lato rappresenta una lista selezionabile, e via dicendo.

Oltre agli ormai consolidati moduli appartenenti alla organizzazione di una GUI bidimensionale, spesso si utilizzano altri artifici quali l'uso di frecce di colori diversi recanti i nomi dei partecipanti, per simulare la posizione su un immaginario desktop (superficie di lavoro) condiviso. L'uso di icone e colori diversi nella lista degli utenti attivi rappresenta solitamente il grado di presenza di queste persone all'interno della collaborazione (attivo, in ascolto, momentaneamente non disponibile).

L'utilizzo di uno spazio condiviso rappresentato attraverso le tre dimensioni presenta l'immediato vantaggio di offrire all'utente una spiccata sensazione di immersività. L'utente non si pone al di fuori dello spazio condiviso (come nel caso di una whiteboard condivisa), ma vi naviga direttamente al suo interno, e ha percezione esatta di come, quando e dove navigano anche altri utenti, ciascuno rappresentato dal proprio 'avatar'. La rappresentazione tridimensionale, dunque, spalanca un repertorio di possibilità vastissimo, che sembrano consentire di potenziare la percezione di collaborazione tra utenti, e di conseguenza di migliorarne gli effetti.

Se tuttavia è vero che - da un lato - la rappresentazione tridimensionale offre una maggiore ricchezza rappresentativa, è anche vero che la sua interpretazione risulta più difficoltosa, sia perché gli utenti non sono abituati ad affrontare rappresentazioni di questo genere, ancora piuttosto nuove, sia perché l'aumento delle sfumature di



**Fig 2-2: ActiveWorlds è un esempio di Collaborative Virtual Environment che consente la cooperazione in uno spazio tridimensionale**

rappresentazione comporta un aumento di funzionalità, non sempre dominabili da parte dell'utente medio.

L'uso della rappresentazione 3D in ambienti collaborativi, dunque, non rappresenta una panacea da applicare dovunque e comunque, per ottenere un maggiore coinvolgimento collaborativo. Si tratta probabilmente di una tecnica adatta ad alcune particolari categorie di applicazioni, che possono trarre un sostanziale giovamento da una maggiore possibilità rappresentativa. Tanti sono gli esempi elencabili, dalle ricostruzioni architettoniche navigabili in modo condiviso, a siti commerciali che propongano oggetti per il cui acquisto è necessaria l'esplorazione attenta dell'oggetto stesso (appartamenti in costruzione, ecc.), alla visita guidata di musei virtuali, a situazioni di e-training che richiedano la conoscenza di percorsi, di oggetti, o di macchinari voluminosi (simulatori, ecc.), da utilizzare in collaborazione con altri operatori o utenti. Rimane tuttavia innegabile come esista un potenziale da analizzare all'interno di queste applicazioni, e come sia necessario esplorarne le caratteristiche e le possibilità per tentare di proporre un modello di fruizione.

A questo scopo gli aspetti da tenere presente sono vari, di natura tecnica e di natura progettuale. Gli aspetti di natura tecnica comprendono:

- la realizzazione di una infrastruttura di comunicazione via rete che consenta di tenere allineati ed in effettiva collaborazione tutti gli utenti
- la realizzazione di un software da affidare a ciascun utente, che consenta effettivamente di implementare e di rappresentare questa collaborazione
- il raggiungimento di prestazioni grafiche adeguate per renderizzare i contenuti visuali e gli aggiornamenti dello spazio condiviso
- l'ottenimento di una certa compatibilità, che consenta a questo software di funzionare in più contesti operativi
- l'usabilità della piattaforma, perché possa essere relativamente facile da utilizzare e da imparare, e sia eventualmente autoesplicativa

Gli aspetti di natura progettuale comprendono il concepimento dell'applicazione e dell'organizzazione dei suoi contenuti (visivi e informativi) in modo che possano essere fruiti in modo ottimale attraverso un paradigma di accesso tridimensionale. Gli aspetti in questo senso sono principalmente due:

- definire come organizzare lo spazio per facilitare la navigazione delle informazioni e l'interazione tra gli utenti ai fini della collaborazione
- definire come organizzare e regolamentare la collaborazione e l'interazione tra gli utenti all'interno della applicazione condivisa.

A questi due aspetti concettuali andrebbe legato un modello da seguire per l'organizzazione dei contenuti. Per quanto riguarda il secondo aspetto concettuale – la collaborazione – dedichiamo un ampio capitolo alla proposta di possibili modelli per la sua descrizione: questo modello viene da noi denominato “*delle Metafore Cooperative*”. A proposito invece del primo aspetto, l'organizzazione spaziale e visuale, occorre trovare delle linee guida che, come per le linee guida delle GUI 2D ormai consolidate, descrivano in modo generale come sia meglio organizzare lo spazio.

Questo modello, derivato principalmente dalla letteratura attuale in proposito, è stato da noi denominato delle *Metafore Visive*.

## 2.5 *Metafore Visive*

Le Metafore Visive<sup>6</sup> sono regole che possono essere applicate in maniera coerente nel progettare la struttura visiva e di navigazione, la topologia e le caratteristiche visive dell'ambiente tridimensionale, allo scopo di favorire o dirigere in un certo modo la collaborazione tra utenti che vi si deve svolgere: con un paragone arditto, una sorta di compendio di *architettura virtuale*.

Il primo aspetto da definire nella progettazione di una metafora visiva consiste nell'adottare una *Metafora Ambientale Complessiva*. Essa riguarda il modo generale scelto per rappresentare lo spazio condiviso, e con quale politica erogare informazioni ai partecipanti.

Le Metafore Ambientali sono legate sostanzialmente al livello di realismo che si intende adottare nell'applicazione e si possono catalogare in:

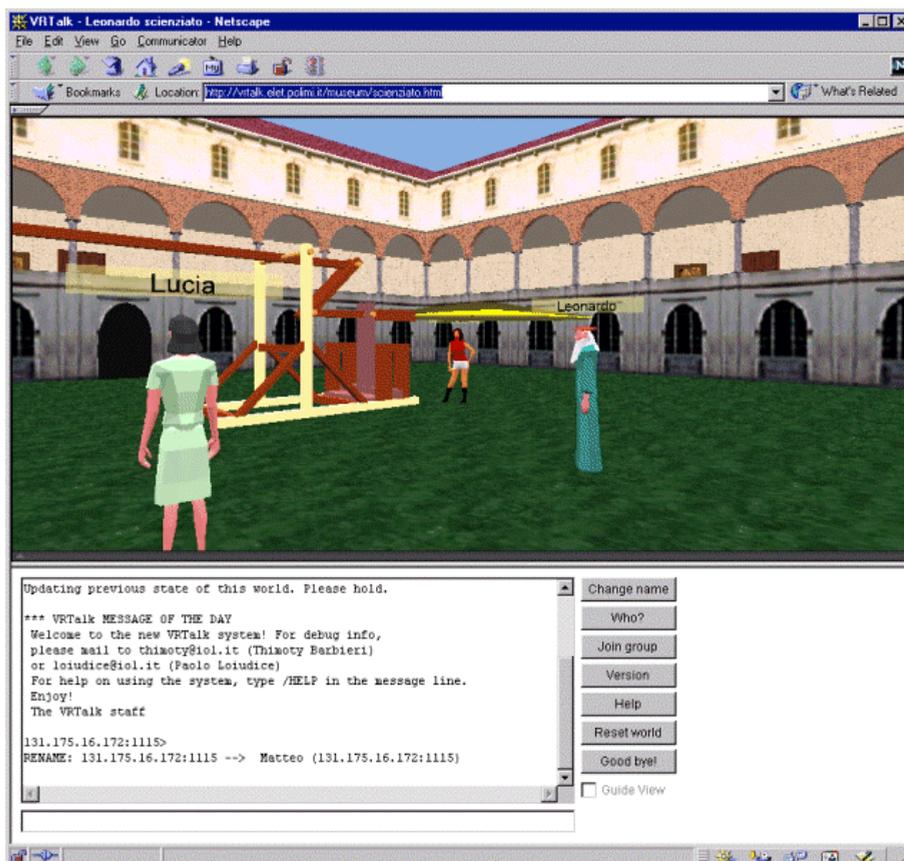
- *Iperrealità*, il cui scopo è rappresentare in dettaglio le complessità che si trovano in un mondo reale, simulandolo quanto più fedelmente possibile.

---

<sup>6</sup> La teoria delle Metafore Visive è tratta da numerose fonti, adattata e modificata per descrivere al meglio il problema di avere delle linee guide per la progettazione visuale di ambienti tridimensionali. Tra le fonti citiamo:

- Bridges, H.A. and Charitos, D., The architectural design in virtual environments, *R. Junge (ed) CAAD Futures'97, Kluwer Academic Publishers, Dordrecht, 1997*
- S. Benford, D. Snowdon, et al. Visualising and Populating the Web: Collaborative Virtual Environments for Browsing, Searching and Inhabiting Workspace, in *proceedings JENC8*
- Cerulli, C. Design of 3D navigable Virtual Exhibition Spaces, in *proceedings Museum&Web 99, New Orleans*
- Charitos, D., Rutherford, P., Ways of aiding navigation within VRML Worlds, *Proceedings of the 6th EuropIA Conference, Edimburgh, 1997*
- Campbell, D., Design in Virtual Environments Using Architectural Metaphors: a HIT La Gallery. *March Dissertation, University of Washington, 1996*
- Charitos, D., Defining existential space in virtual environments, *Proceedings of Virtual Reality Worlds '96, February '96, IDG Magazines, Stuttgart, 1996*

- *Realtà selettive*, rappresentazioni semplificate dei mondi fisici, in cui alcuni particolari aspetti dell'ambiente sono trattati con un maggiore livello di realismo, mentre altri sono trasformati e adattati (semplificandoli o variandone le caratteristiche fisiche), altri ancora sono del tutto ignorati, perché pertinenti ad informazioni non necessarie al contesto dell'applicazione.



**Fig. 2-3:** L'applicazione collaborativa 3D "Leonardo Virtuale" adotta una Metafora Ambientale di tipo Selettivo. Viene rappresentata una realtà semplificata (il cortile ha pochi dettagli, e mancano le scale).

- *Astrazioni*, che rappresentano particolari ambienti virtuali, in cui si rappresentano informazioni astratte derivate da ambienti reali di elevata complessità (*astrazione sui contenuti*), oppure informazioni tali da non avere una effettiva rappresentazione fisica (*astrazione strutturale*).

Un esempio di Realtà Selettiva si può trovare nella nostra realizzazione dell'ambiente "Leonardo Virtuale"<sup>7</sup>, che consente di navigare in una rappresentazione del cortile interno del Museo della Scienza e della Tecnica, e di entrare in varie stanze, contenente modelli delle macchine di Leonardo da Vinci. In particolare, per salire ai piani superiori occorre "volare", dato che nel modello virtuale (al contrario che nel museo reale) non esistono scale.

Un esempio di astrazione strutturale si può trovare in motori di ricerca sperimentale che modellano la struttura delle informazioni presente nel Web utilizzando delle metafore tridimensionali<sup>8</sup>. In questo caso l'informazione rappresentata non ha una effettiva rappresentazione fisica (pagine web e informazioni al loro interno, e relativi collegamenti ipertestuali), ma la necessità di visualizzazione costringe ad adottare una astrazione strutturale.

Definita una metafora ambientale adatta, occorre poi stabilire dei *pattern spaziali* per la presentazione dell'informazione: essi servono a stabilire quali sono e come sono rappresentati i generici elementi preposti a definire lo spazio nell'ambiente virtuale, associando una forma (elemento o pattern spaziale) per ciascuna delle funzioni individuate.

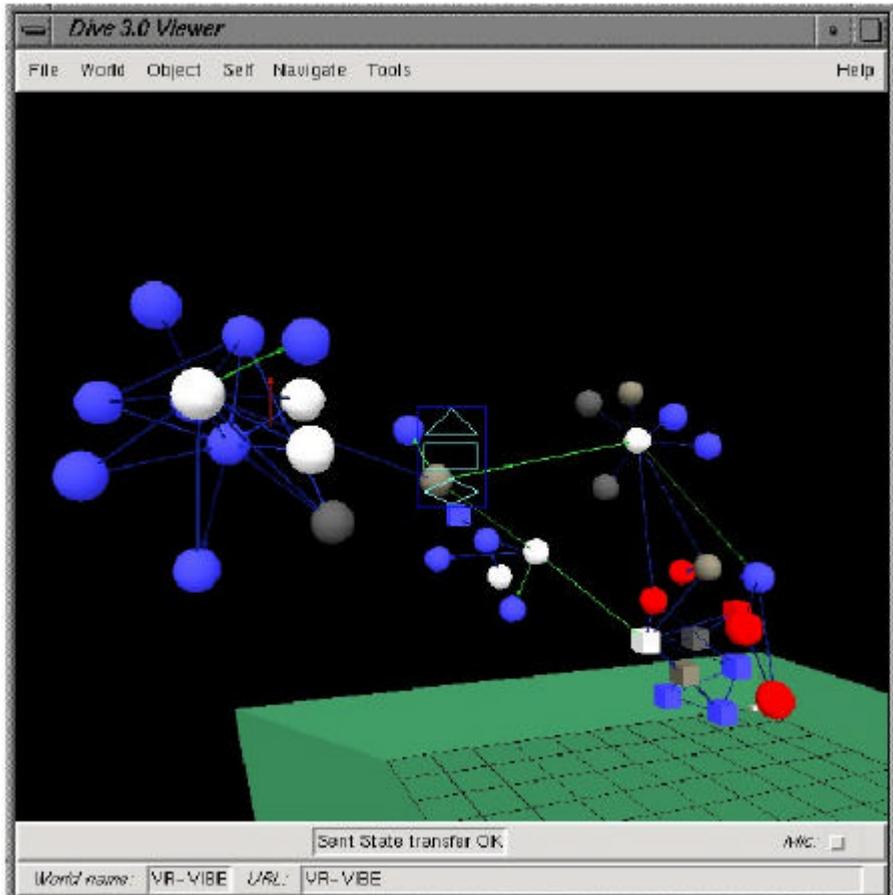
Le funzioni assolvibili dagli elementi spaziali si dividono sostanzialmente in tre categorie:

- *Scenari*: sono elementi spaziali principalmente vuoti, che sono usati per definire i confini della possibile navigazione
- *Allestimenti*: sono oggetti spaziali solidi che vengono usati per popolare gli scenari, consentendo di distinguere uno scenario dall'altro
- *Ipercollegamenti*: sono oggetti funzionali speciali, che consentono di spostare l'utente da una locazione (scenario) all'altra.

---

<sup>7</sup> Paolini P., Barbieri T., et al. – "Visiting a Museum Together: how to share a visit to a virtual world", in *Proceedings Museums&Web '99, New Orleans (USA), Marzo 99*, pg. 27 e segg.

<sup>8</sup> S. Benford, D. Snowdon, et al. *Visualising and Populating the Web: Collaborative Virtual Environments for Browsing, Searching and Inhabiting Workspace*, in *proceedings JENC8*



**Fig. 2-4:** In questo ambiente viene rappresentata la struttura di un sito Web. E' possibile navigarlo in modo collaborativo partendo da una rappresentazione astratta strutturale.

Un *pattern spaziale* ideato per rappresentare uno *Scenario* deve definire le seguenti caratteristiche:

- *Locazione:* sono gli spazi in cui viene svolta una attività collaborativa. Una locazione viene definita dai suoi confini geometrici e deve essere progettata in modo da consentire all'utente di riconoscere un *interno* ed un *esterno* rispetto alla locazione, per consentirgli di capire quando ne *entra* e quando ne *esce*.
- *Percorsi:* sono spazi che con il loro aspetto nello spazio suggeriscono un movimento da un punto di partenza ad uno di arrivo (es. una strada).
- *Domini:* sono aree spaziali non strutturate in cui le *locazioni* e i *percorsi* possono essere raggruppati. I domini sono percepiti attraverso un processo di astrazione mentale dopo la navigazione in tutti i percorsi e tutte le locazioni

che li compongono (es. si distinguono le zone residenziali dalle zone industriali esplorando una città).

- *Soglie*: sono spazi di transizione o di intersezione tra qualsiasi altro elemento spaziale

A ciascuna di queste caratteristiche deve venire assegnata in modo coerente una rappresentazione geometrica (ad esempio, si decide di rappresentare i percorsi come strade, le soglie come steccati, le locazioni come case).

Il *pattern spaziale* che rappresenta un *Allestimento* viene invece definito attraverso le seguenti caratteristiche:

- *Oggetti Guida*: sono oggetti che in qualsiasi modo limitano o definiscono lo spazio di collaborazione (mobili, luci, scale...)
- *Punti di riferimento (landmarks)*: sono oggetti dai caratteri nettamente distinti che servono come punto di riferimento all'interno dell'ambiente. La conoscenza dei punti di riferimento è molto importante per orientare l'utente all'interno del mondo virtuale. La connessione mentale tra più punti riconosciuti come di riferimento forma un *percorso*. Imparare un insieme di *percorsi* dà all'utente la conoscenza della configurazione topologica e strutturale del mondo. Ciascun utente in generale si crea un proprio insieme di *percorsi* ed ha il proprio grado di conoscenza del mondo, avendone dunque una percezione leggermente differente.

La terza tipologia di *pattern spaziale* riguarda la definizione degli *Ipercollegamenti*. Nel mondo reale, gli ambienti vengono esplorati unicamente per movimento continuo nello spazio e nel tempo. Gli ipercollegamenti, invece, consentono di saltare istantaneamente tra spazio (e quindi porzioni di dati) non adiacenti. Ce ne sono di due diversi tipi, eventualmente rappresentabili in modo diverso, per evitare di confondere l'utente:

- *Contigui*: connettono ambienti contigui tra loro all'interno di un mondo, ed hanno una forte correlazione dal punto di vista del contenuto informativo. Questi collegamenti sono usati per scomporre l'ambiente (e l'informazione interconnessa) in unità più piccole e comprensibili, per offrire una navigazione più agevole (per es. una porta di una stanza di un palazzo)
- *Collegati*: connettono parti che non sono spazialmente adiacenti, ma contengono informazioni logicamente correlate (per es. un fascio di luce teletrasportante).

Un'ultima proprietà delle Metafore Visive è la tipologia di *Riorganizzazione Spaziale* che vi si vuole assegnare. La Riorganizzazione Spaziale è la capacità di rielaborare i pattern spaziali del mondo virtuale in relazione a stimoli esterni provenienti dall'utente (informazioni relative alle sue caratteristiche, quali sesso, età, preferenze artistiche; le sue azioni e scelte all'interno dell'ambiente durante la navigazione, ecc.).

Occorre stabilire in anticipo se la Metafora Visiva è legata ad una serie di pattern spaziali fissi oppure dinamici, cioè se è consentita durante la navigazione una riorganizzazione dello spazio condiviso.

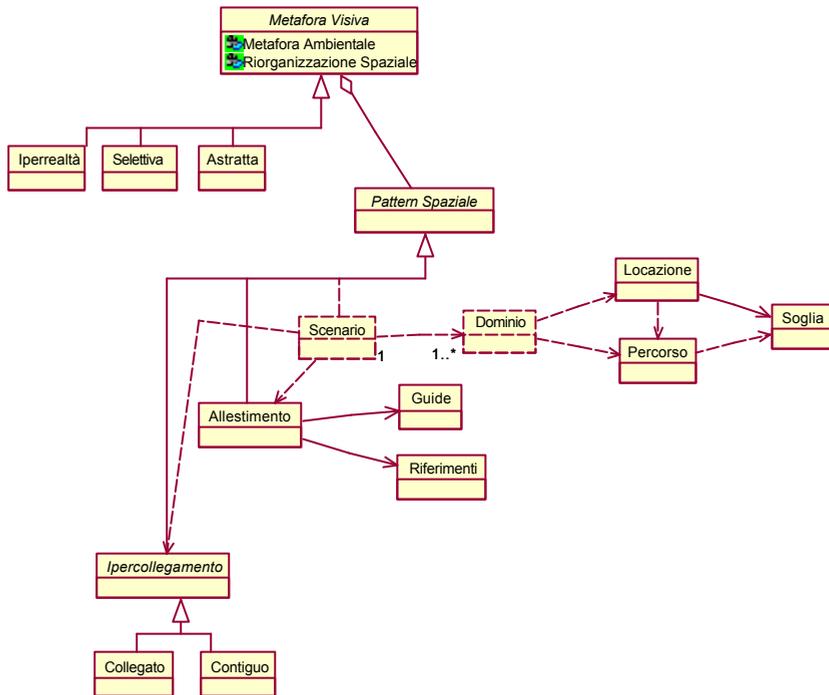
Bisogna tuttavia tenere presente che la possibilità di riorganizzare lo spazio in risposta ai comportamenti utente, sebbene affascinante, è in antitesi con il concetto di collaborazione: infatti la riorganizzazione distrugge tutta l'esperienza cognitiva fatta in termini di punti di riferimento e percorsi che li interconnettono delle precedenti navigazioni e nelle navigazioni di altri utenti con cui si collabora, e questo va a detrimento del livello di consapevolezza dello stato condiviso.

Un esempio applicativo di Metafora Visiva è stato tentato modellando un'applicazione prototipo denominata "Il Museo Virtuale dell'Informatica Italiana"<sup>9</sup>. Lo scopo di questo museo era presentare alcuni contenuti informativi riguardo alla storia dello sviluppo dell'informatica in ambito accademico ed industriale, attraverso fotografie, interviste, filmati e giochi interattivi tridimensionali.

La struttura di accesso tridimensionale dell'applicazione realizzata, qui descritta brevemente in termini di metafore visive, utilizza una astrazione concettuale, usando una serie di locazioni a forma di "piattaforme volanti". Ciascuna piattaforma volante utilizza la stessa geometria, ma viene identificata da un oggetto che serve come punto di riferimento. La navigazione tra una piattaforma e l'altra avviene attraverso ipercollegamenti *collegati*, rappresentati in modo da suggerire un fantascientifico teletrasporto.



**Fig 2-5:** Una analisi di ambiente virtuale in termini di Metafore Visive. Le locazioni nel "Museo Virtuale dell'Informatica Italiana" sono rappresentate da piattaforme, utilizzando una metafora ambientale di astrazione concettuale. Ogni piattaforma utilizza come allestimento un landmark per consentirne il riconoscimento dalle altre.



**Fig 2-6: Diagramma che descrive le relazioni concettuali tra le definizioni del modello delle *Metafore Visive*. Una *Metafora Visiva* è una collezione di più *pattern spaziali*, scelti per rappresentare volta per volta *Scenari*, *Allestimenti* e *Ipercollegamenti*.**

## 2.6 Architettura Generale di un Sistema Collaborativo Multicanale

Per la realizzazione di sistemi collaborativi che usino rappresentazioni tridimensionali, si è detto come i problemi da affrontare siano sia di natura tecnica che concettuale. Sul piano concettuale, sulle *Metafore Visive* s'è detto, mentre si tratterà il problema di presentare un modello di collaborazione nei capitoli successivi.

Sul piano tecnico, occorre individuare alcuni problemi tipici di questo tipo di sistemi collaborativi, e offrire delle possibili soluzioni, inquadrandole in un contesto tecnico

<sup>9</sup> Barbieri T., Garzotto F., et al – “From Dust to Stardust: a Collaborative Virtual Computer Science Museum”, in *Proceedings Short Papers Ichim 01, Milano, Italy, September 2001*

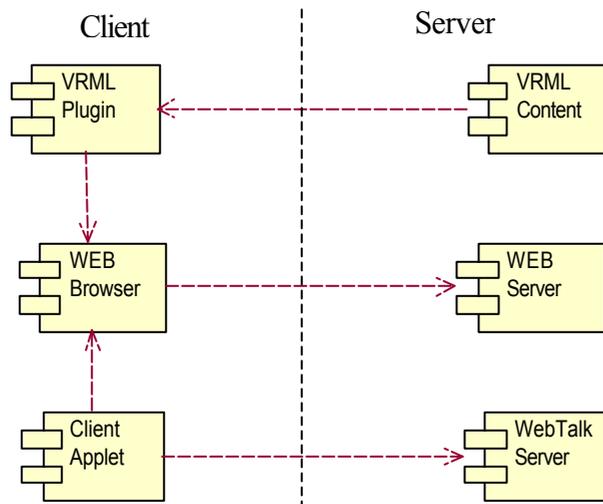
generale. Un primo tentativo di implementazione è stato eseguito utilizzando un'architettura tecnologica da noi denominata WebTalk-I. Questa architettura è stata utilizzata innanzitutto agli inizi della ricerca, quando occorreva ancora valutare e comprendere i meccanismi fondamentali della collaborazione negli spazi tridimensionali, ed i modi migliori di organizzare lo spazio 3D per supportarla. Dall'esperienza raccolta con questa architettura, ne è stata ideata una seconda, denominata WebTalk-II, con l'intento di rispondere ai principali quesiti sorti dall'analisi del problema effettuata mediante la prima architettura.

### 2.6.1 WebTalk-I

L'architettura WebTalk-I<sup>10</sup>, sviluppata tra il 1997 e il 1999, ha come idea principale la realizzazione di un'applicazione di collaborazione 3D che potesse girare esclusivamente all'interno di un browser Internet.

Il risultato è stato raggiunto utilizzando un plug-in di visualizzazione e interazione utente per lo standard tridimensionale VRML 97, utilizzato in combinazione con una Applet Java per la raccolta lato client degli eventi e il loro scambio verso un server, preposto alla gestione dello stato condiviso, e anch'esso scritto in Java.

La comunicazione tra il plug-in VRML e l'applet Java è stata resa possibile da una API di programmazione denominata EAI (External Authoring Interface), che consente ad un'applicazione Java di prelevare dei puntatori ai nodi geometrici di una scena 3D caricata in un embed di una pagina del browser.



**Fig 2-7: Componenti dell'architettura WebTalk-I**

<sup>10</sup> Barbieri T. – “Networked Virtual Environments for the Web: The WebTalk-I and WebTalk-II Architectures”, in *Proceedings IEEE for Computer Multimedia & Expo 2000 (ICME)*, New York, USA, July 2000

Il server WebTalk si occupa della ridistribuzione dei messaggi a tutti gli applet client che si connettono ad una scena virtuale. I messaggi possono essere di varia natura:

- Messaggi di chat: con i quali gli utenti possono dialogare on-line
- Messaggi aggiornamento del mondo: che possono riguardare gli spostamenti degli avatar degli utenti all'interno del mondo virtuale oppure manipolazione di oggetti condivisi
- Messaggi di sistema: utilizzati per funzioni accessorie del sistema come il cambio di nome di un utente, il collegamento e lo scollegamento ecc..

Questi messaggi sono gestiti utilizzando un particolare protocollo applicativo, costituito da un comando e da parametri il cui significato posizionale dipende dal comando effettivamente inviato. Il server elabora il comando ricevuto e aggiorna la propria immagine dello stato condiviso in memoria, ridistribuendone le variazioni a tutti i client di competenza. Il server di WebTalk-I gestisce anche alcune primitive regole di collaborazione, che stabiliscono come gli utenti possono interagire tra di loro, ed in quali contesti.

### **2.6.2 WebTalk-II**

Lo sviluppo di applicazioni basate sull'architettura di WebTalk-I ha consentito di comprendere meglio la natura delle applicazioni collaborative tridimensionali, e di delineare meglio alcune problematiche tipiche:

- *Portabilità ed evolvibilità*: la soluzione legata ad un plug-in (CosmoPlayer, piuttosto che Cortona o Shockwave 3D, o altri ancora), lega l'applicazione ad una particolare piattaforma se il plug-in è implementato solo per essa (Cortona<sup>11</sup>, ad esempio, è implementato per PC e Mac), e lega l'applicazione stessa alla nascita, fortuna e morte del plug-in utilizzato. Questo è stato uno dei problemi della piattaforma precedente: il plug-in utilizzato, Cosmo Player, non ha più continuato il proprio sviluppo, e non era disponibile per altre piattaforme. Una soluzione da noi proposta è realizzare il client grafico attraverso un sistema portabile, ad esempio mediante le API grafiche Sun Java 3D.
- *Prestazioni*: l'utilizzo di plug-in oppure di API grafiche di alto livello (quali ad esempio Java 3D) costringe ad accettare compromessi sulle prestazioni nella resa grafica dell'ambiente. Questo a causa del fatto che è il codice del plug-in o della API a fissare le priorità e le modalità di rendering, lasciando poca libertà nell'ottimizzare le prestazioni di calcolo verso alcuni aspetti ben mirati. Una soluzione potrebbe essere quella di scrivere un microkernel di rendering grafico direttamente utilizzando un linguaggio accelerato di livello

---

<sup>11</sup> Cortona VRML Player: <http://www.parallelgraphics.com>

programmatico medio-basso, come ad esempio C con OpenGL, ormai accelerato in hardware nella maggior parte delle schede grafiche di medio e alto livello.

- *Coordinamento*: la chiave di volta e lo scopo di questo genere di applicazioni è favorire la collaborazione, e dirigerla affinché possa essere raggiunto un determinato scopo operativo. Tuttavia, la maggior parte delle applicazioni disponibili in questo campo trascurano di regolamentare i rapporti e le modalità di collaborazione tra gli utenti. Risulta invece più proficuo avere la possibilità di descrivere come la collaborazione deve avvenire, guidandola quindi all'ottenimento di un determinato fine. A questo scopo abbiamo prima sviluppato un modello concettuale, le *metafore cooperative*, e poi ne abbiamo tentato diverse implementazioni, a vari livelli.
- *Scalabilità*: Il supporto per i grandi gruppi ed i grandi ambienti costituisce un grosso problema tecnologico per questa categoria di sistemi. Il mantenimento dello stato condiviso lato server oppure su ciascuna istanza locale, il rendering in tempo reale dei movimenti e dei cambiamenti di stato, ed altre problematiche che richiedono sempre maggiori risorse di rete e computazionali al crescere del numero degli utenti, richiedono uno studio architettonico attento. A questo problema, si propone una architettura a elementi clusterizzabili, ed un server ottimizzato che utilizza alcune tecniche miste per raggiungere maggiori prestazioni, quali interest management, dead reckoning, ecc.
- *Contenuti*: la raccolta e la realizzazione dei contenuti, fine ultimo della collaborazione tra utenti, richiede l'approntamento di siti web convenzionali, alle cui componenti si può raggiungere dopo aver collaborato all'interno di uno spazio tridimensionale. Tuttavia questo implica un doppio sforzo di progettazione: progettare la struttura dei contenuti bidimensionali, e poi progettare nuovamente l'organizzazione del mondo tridimensionale. Occorrerebbe garantire una certa consistenza tra i contenuti a cui arrivare mediante la collaborazione, e la struttura tridimensionale di accesso che vi si cala sopra. Sulla gestione dei contenuti, proponiamo una mappatura di una modellazione concettuale delle informazioni ipermediali e delle relative strutture di navigazioni bidimensionali, su paradigmi di accesso e navigazioni tridimensionali, mediante la definizione di uno schema XML.
- *Desolazione*: uno dei problemi riscontrati dalle prime applicazioni collaborative 3D, è dato dalla loro giovinezza e dalla scarsa conoscenza che ne ha il pubblico di Internet. Il risultato è che, qualora si desideri realizzare un'applicazione dedicata al vasto pubblico, e non diretta ad un target ben definito di persone, occorre mettere in conto la scarsa dimestichezza con questo tipo di sistemi da parte degli utenti medi. Inoltre, nei sistemi collaborativi l'interesse sussiste qualora vi sia della collaborazione da operare. Un ambiente vuoto è difficile da interpretare (non si riesce ad intuire immediatamente in che direzioni esplorarlo e quali funzionalità offre), e poco

interessante da navigare. Una soluzione da noi proposta è la creazione di entità agenti e entità puppet, pilotate algoritmicamente, che possano assolvere a due funzioni: la prima, di ausilio e dimostrazione sulle modalità di utilizzo del sistema collaborativo, e la seconda per creare popolazione e trattenere i primi utenti al loro ingresso al sistema, in modo da far poter offrire sempre un livello di collaborazione iniziale.

- *Multicanalità*: Questa tipologia di sistemi collaborativi impone delle barriere di ingresso consistenti in termini di risorse necessarie per accedervi: processore grafico, connessione di rete a banda larga, postazione di lavoro fissa sono solo tre dei principali requisiti necessari. Si può pensare di allargare la collaborazione anche in ambiti meno ristretti, ad esempio verso utenti che sono collegati alla rete mediante un link radio mobile. Come realizzare questo incontro tra sistemi di collegamento tanto diversi, e quali ne siano i risultati, è stato oggetto dei nostri primi studi nell'ambito della collaborazione multicanale.

I problemi tipici affrontati dai gruppi di ricerca che si occupano di CVE e CSCW sono le prestazioni in ambito grafico e nella scalabilità al supporto di grandi gruppi. La presente ricerca, prima fra tutte, tenta di dare una panoramica generale di tutte le questioni (non solo dunque quelle che entrano nel merito prestazionale, ma anche e soprattutto quelle che richiedono una modellazione concettuale del problema a priori) che occorre considerare per realizzare un sistema collaborativo di successo, offrendo soluzioni innovative, e ponendo problemi che non sono in generale sollevati da parte di altri gruppi che lavorano nel settore, quali ad esempio il problema di regolamentare la collaborazione, di gestire i contenuti coordinando la progettazione concettuale con la sua struttura di accesso tridimensionale, e di valutare l'impatto dell'accesso attraverso canali di natura diversa alle stesse informazioni condivise, esplorando le frontiere della multicanalità.

Sotto il nome progettuale "WebTalk-II" indichiamo un generico framework di progetto che si propone di affrontare le problematiche sopra descritte, mediante la creazione di un'architettura generale, che è stata di base per la nostra ricerca fino a questo punto e per le sue successive eventuali soluzioni.

Il framework generale del sistema collaborativo da noi proposto si compone di due grandi blocchi: quello preposto alla gestione dello stato condiviso, e quello preposto alla rappresentazione locale dello stato condiviso presso l'utente.

Il gestore dello stato condiviso si immagina utilizzabile da una qualsiasi delle versioni della rappresentazione locale. Il suo compito è di tenere traccia di una copia dello stato istantaneo dello spazio condiviso, da fornire a tutti i partecipanti, effettuando eventualmente operazioni di ottimizzazione, e filtraggio. Tra le operazioni possibili, vi è anche quella di regolare e governare la conversazione, l'interazione e la collaborazione tra gli utenti, attraverso una opportuna serie di regole implementate in stretto contatto con la gestione dello stato condiviso. Altri moduli riguardano la gestione degli Agenti di

assistenza all'utente, e di collaborazione automatizzata. Di particolare rilievo uno strato denominato "Wireless Gateway", il quale deve occuparsi di effettuare le semplificazioni ed i filtri necessari per presentare una versione semplice ed una connessione di rete compatibile con periferiche e canali di tipo diverso da quelli utilizzati dai client di rappresentazione tridimensionale. Periferiche di questo tipo che si sono immaginate sono cellulari dotati di un browser WAP, oppure palmari (PDA), dotati di uno speciale client semplificato, oppure ancora programmi di Instant Messaging da mantenere sul desktop degli utenti, sullo stile di AIM, ICQ, o MSN.

Un ultimo compito del gestore dello stato, è analizzare la struttura iniziale dello stato condiviso mediante il caricamento, l'inizializzazione e l'invio della presentazione verso i client dell'applicazione collaborativa fornita. Questa struttura viene progettata e generata attraverso un modulo denominato "configuratore", che si occupa di guidare nella progettazione l'autore dell'applicazione collaborativa destinata a girare utilizzando il framework. Il risultato della progettazione è uno schema che spiega quale è l'effettiva geometria del mondo tridimensionali, quali le possibilità di interazione con gli oggetti che lo compongono, e quali le possibili modalità di collaborazione tra gli utenti. Queste regole di configurazione sono lette in fase di caricamento dal gestore dello stato condiviso, e eseguite a runtime durante l'interazione con i client partecipanti.

La struttura del framework per quanto riguarda la rappresentazione tridimensionale dello spazio condiviso si fonda essenzialmente su due concetti astratti: lo Scene Builder e la GUI bidimensionale.

Lo Scene Builder è una porzione di software il cui compito è interpretare le informazioni strutturali, geometriche e navigazionali specificate dal Configuratore e diramate dal gestore dello stato condiviso, e comporre una scena in un modo efficace per la tecnologia di rendering real-time a video selezionata. La GUI Bidimensionale rappresenta tutta la parte di grafica bidimensionale necessaria alla normale interazione con l'utente, quale ad esempio la finestra di chat, i pulsanti per attivare o disattivare le funzioni, la lista degli utenti presenti in una determinata zona dello spazio applicativo, e via dicendo. Poiché esistono più tecnologie possibili sia per il rendering 3D che per la rappresentazione della GUI, il framework modella come astratti questi due concetti, proponendo delle possibile tecnologie per la loro implementazione.

Così, combinazioni possibili risultano essere una applet Java con GUI Swing che interagisce con un plug-in VRML (come nell'architettura WebTalk-I), piuttosto che Swing con un renderer scritto in Java3D, oppure ancora l'uso di OpenGL con una GUI scritta con le Foundation Classes di un sistema operativo nativo.

In tutti i casi, sarà inoltre necessario uno strato di rete che interagisca strettamente con lo Scene Builder e con la GUI, per consentire l'invio in rete e la ricezione da parte del gestore dello stato condiviso sia di messaggi di comunicazione, che di messaggi atti ad aggiornare la rappresentazione grafica dello stato condiviso.

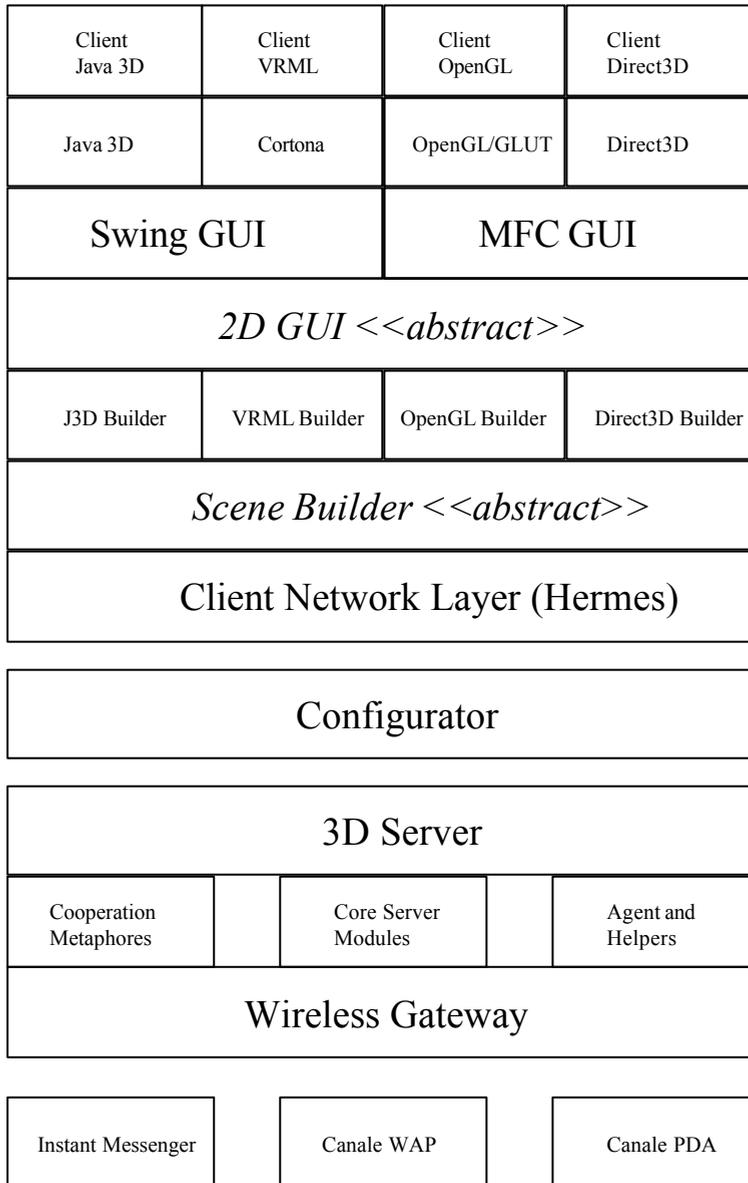
Nel corso della nostra ricerca, non sono stati effettivamente implementati tutti i moduli concettuali definiti in questa analisi generale, né sussiste, allo stato dei fatti una perfetta interoperabilità tra tutti i moduli sviluppati.

Tuttavia, sono stati completati parecchi moduli, che sono stati utilizzati per provare la validità concettuale e pratica delle soluzioni ipotizzate. In particolare, la ricerca e lo sviluppo si sono concentrati su realizzazioni alternative della gestione dello stato condiviso, sulla realizzazione di un configuratore, su quella di un client implementato in Java3D (dopo aver sviluppato un client basato su plug-in VRML nell'architettura precedente).

Alcuni esperimenti prestazionali sono stati eseguiti con una prima versione di un microkernel di navigazione implementato direttamente in OpenGL, al fine di valutare come l'implementazione di alcune politiche di ottimizzazione del rendering potessero avere un effetto benefico sulla navigabilità e quindi sull'aumento della possibilità di collaborare.

Un'altra fase della ricerca e sviluppo ha riguardato la realizzazione di un modulo di gestione degli agenti automatici ad ausilio della navigazione e della collaborazione, e uno studio dell'utilizzo di questo sistema in un contesto multicanale relativamente arduo, che prevedeva la collaborazione con terminali mobili molto limitati (cellulari con WAP) e terminali mobili a capacità superiori (PDA con sistema operativo PalmOS e WindowsCE).

Alla base della trattazione seguente, dunque, rimane sempre lo schema generale del framework collaborativo, che rappresenta un inquadramento complessivo, un approccio globale alla completa struttura che, secondo la nostra visione, un "sistema collaborativo totale" dovrebbe avere.



**Fig. 2-8: la struttura generale del framework di collaborazione oggetto della ricerca.**

## 3 Le Metafore Cooperative

### 3.1 Il concetto di metafora cooperativa

In molti casi, gli utenti accedono allo spazio condiviso rappresentato da un sito web in modo solitario, senza nessuna consapevolezza di quello che gli altri utenti stanno facendo nello stesso momento. In un vero sistema collaborativo, invece, i vari utenti devono poter raggiungere insieme un obiettivo, accedendo contemporaneamente allo spazio condiviso. Gli obiettivi di una attività collaborativa possono essere tra i più vari: insegnamento e apprendimento in un ambito di e-learning, shopping in un sito di e-commerce che offre una rappresentazione virtuale dei prodotti, progettazione collaborativa di macchine o ambienti, supervisione e controllo di operazioni svolte in ambienti che nella realtà sono pericolosi, consulenza, ecc.

Il concetto di collaborazione viene associato con molti comportamenti, ma è possibile individuare due estremi di uno spettro molto variegato di situazioni:

- gli utenti che collaborano sono pari, con le stesse conoscenze e le medesime possibilità
- uno o più degli utenti controllano la situazione, con conoscenze o possibilità operative superiori a quelle degli altri, come ad esempio per degli istruttori con degli allievi, dei commessi di negozio rispetto a degli acquirenti, delle guide turistiche con dei visitatori, ecc.

Una delle principali osservazione da cui parte l'esigenza di trovare un modello delle modalità di collaborare negli spazi virtuali, è che nel mondo reale la maggior parte delle attività si svolgono, ove possibile, in modo collaborativo. La cooperazione mentre si cerca di ottenere qualcosa molto spesso risulta più interessante, più coinvolgente, più divertente e molto più efficace. Questa osservazione iniziale si collega con il fatto che gli ambienti virtuali condivisi esistono ormai da alcuni anni. Inizialmente solo in forma testuale (MUD e MOO), consentendo agli utenti di scambiarsi messaggi per iscritti, e nelle loro successive evoluzioni con l'introduzione di grafica tridimensionale (Blaxxun, Active Worlds<sup>12</sup>), audio, ed altre funzionalità avanzate. In questi mondi virtuali, gli utenti sono rappresentati come oggetti virtuali solitamente antropomorfi, denominati "avatar"; gli utenti possono muoversi all'interno della geometria tridimensionale, interagire con il mondo virtuale e gli oggetti che compongono, parlare insieme, ecc.

Questi ambienti consentono già un elevato livello di interazione tra gli utenti, ma mancano di alcune caratteristiche che, a nostro avviso, sono cruciali dovunque lo scopo della collaborazione non è il semplice incontro tra le persone, ma una esplorazione guidata e una ricerca finalizzata dell'informazione.

---

<sup>12</sup> <http://www.activeworlds.com>

Nell'uso attuale dei mondi virtuali cooperativi, infatti, l'interazione tra gli utenti sembra essere l'obiettivo, piuttosto che il mezzo per ottenere qualcosa o trovare un'informazione. Gli utenti si divertono parlando e interagendo tra loro, ma non cercano di ottenere uno specifico risultato da questa esperienza. Gli utenti di un sistema collaborativo rivolto a supportare un gruppo di lavoro (CSCW), invece, hanno un obiettivo comune, quello di trovare o generare nuova informazione.

Si consideri l'esempio di un museo virtuale condiviso: l'obiettivo degli utenti è visitare il museo e trovare informazioni interessanti al suo interno, di comprenderne meglio la natura e le caratteristiche. La quantità di informazione che viene scambiata dagli utenti, o quella resa disponibile nei mondi virtuali attuali, è relativamente piccola e organizzata in modo semplice. I veri siti web museali, di per contro, diventano sempre più complessi, e forniscono un grande volume di informazioni, spesso su argomenti specifici e specialistici.

La cooperazione che avviene nel modo attuale di utilizzare gli ambienti virtuali, è sparsa e mal guidata, nel senso che gli utenti sono liberi fondamentalmente di comportarsi come desiderano, con pochissimi vincoli e nessun comportamento predeterminato cui attenersi. Le nostre esperienze nell'uso dei mondi virtuali, invece, dimostrano che la collaborazione deve essere organizzata e seguire delle precise linee guida per essere veramente efficace.

La nostra percezione delle interazioni nei mondi virtuali si è sviluppata soprattutto mediante la realizzazione di applicazioni museali, il cui scopo era l'erogazione di informazioni e conoscenza, ma in generale queste esperienze sono applicabili a una serie di casi simili, quali attività di e-shopping, di consulenza, e di progettazione collaborativa, che avvengono in un ambiente tridimensionale il cui scopo è fornire accesso ad un corpo complesso di informazioni memorizzato sotto forma di contenuti ipermediali e ipertestuali. Dallo studio di queste applicazioni e dalle realizzazioni prototipali sono emersi vari punti:

- L'accesso a un sito web tradizionale può essere reso più interessante, coinvolgente ed efficiente consentendo un certo grado di collaborazione tra gli utenti che accedono ad un sito nel medesimo momento
- La collaborazione deve seguire delle linee guida ben organizzate, per poter essere efficace e raggiungere un determinato obiettivo (l'apprendimento, l'acquisto, ecc.).
- Un mondo tridimensionale virtuale, con degli avatar a rappresentare la "presenza" degli utenti, può essere una rappresentazione interessante per una collaborazione efficace e coinvolgente.
- L'informazione rappresentata direttamente nello spazio di collaborazione (il mondo tridimensionale) non può riprodurre tutte le informazioni offerte dal sito Web a cui si riferisce, e che è l'oggetto della collaborazione. Una quantità eccessiva di informazione, infatti, rende disordinato lo spazio di collaborazione, rendendolo meno utile ed efficiente.
- Lo spazio di collaborazione deve offrire una quantità sufficiente di informazioni per consentire all'utente di trovare l'oggetto della propria ricerca sul sito web collegato; una volta che le informazioni di interesse sono state

localizzate, ci devono essere dei punti di passaggio che consentono all'utente di passare dall'ambiente tridimensionale collaborativo alla porzione richiesta del sito Web.

Quando gli utenti collaborano all'interno di sistemi pensati con le finalità sopra elencate, devono poter seguire delle modalità di interazione che definiamo "*metafore cooperative*". Questa interazione si basa sul concetto di *percezione dello stato condiviso*, già trattato in precedenza: è questa percezione che, in una sorta di autoanello cognitivo, innesca un processo collaborativo che alimenta ulteriormente la necessità e la possibilità di collaborare. L'utilizzo di alcune possibilità della rappresentazione tridimensionale permette di immaginare modi di rappresentare queste interazioni, queste forme di collaborazione, queste *metafore collaborative*, che siano nuovi e espressivamente più potenti.

Tuttavia, il problema di definire la forma corretta di collaborazione è spesso legata al tipo di applicazione che si vuole realizzare (cioè al problema specifico che si affronta, e al modo con cui visivamente lo si presenta): in questo senso non tutte le forme di collaborazione sono sempre necessarie per eseguire un'attività all'interno di un dato ambiente.

Proprio per questo motivo è importante capire, prima di realizzare un sistema collaborativo che si appoggi sulle tre dimensioni, in che modo le informazioni sono presentate in tre dimensioni (adottando il modello guida delle *metafore visive*, presentato nei capitoli precedenti), e determinare precisamente quali sono le regole secondo cui gli utenti sono in grado di collaborare gli uni con gli altri, e con l'ambiente, per aumentare la propria percezione dello stato condiviso durante la loro attività.

Le regole scelte per definire questa collaborazione formano nel loro complesso le *metafore cooperative*.

Nella nostra visione, una *metafora cooperativa* è un insieme di *regole base* che descrivono le diverse modalità di interazione tra gli utenti, e tra gli utenti e l'ambiente in cui si trovano. Queste regole comprendono vari aspetti, come il modo con cui gli utenti si raccolgono in gruppi per parlare insieme oppure navigare nello spazio virtuale, o come la visualizzazione dello stato degli oggetti e degli avatar deve essere eseguita. Prendendo delle precise decisioni riguardo a tutti questi aspetti, seguendo la struttura delle *metafore cooperative* per modellare l'aspetto dinamico (di fruizione) del mondo virtuale – esattamente come si può seguire il modello delle *metafore visive* per definirne l'aspetto strutturale – è possibile definire una *descrizione globale* (chiamata metafora) delle diverse possibili forme di collaborazione tra gli utenti nel sistema.

L'esperienza fatta nella progettazione di applicazioni collaborative ha dimostrato che sebbene le metafore di cooperazione fondamentali siano le stesse per tutte le applicazioni, il modo di realizzarle effettivamente o di utilizzarle in modo combinato cambia da applicazione ad applicazione. Il ruolo del progettista di un ambiente di collaborazione, pertanto, non si limita solamente alla raccolta dei contenuti e al disegno delle geometrie tridimensionali che rappresentano l'ambiente, ma comprende anche

l'ideazione di una metodologia coerente di interazione mediante l'uso di metafore cooperative adatte alle diverse situazioni applicative; i progettisti, in altre parole, devono descrivere le regole di interazione tra gli utenti all'interno dell'applicazione.

Un generico framework che supporti la collaborazione su sistemi di rete, quindi, più che fornire un insieme di modalità prefissate di collaborazione, deve dare al progettista una vasta rosa di possibilità che possono essere scelte e regolate sulle particolari esigenze applicative. Questo aspetto manca in tutti gli strumenti e le piattaforme di comunità virtuali oggi disponibile, e questo è il motivo per cui uno degli aspetti di cui ci siamo occupati è quello di definire un modello in cui delle potenti metafore cooperative possano essere combinate e regolate alla bisogna.

### **3.2 Il modello base**

A grandi linee, si può affermare che la collaborazione possa essere rappresentata con i due paradigmi di visualizzazione corrispondenti al contesto bidimensionale e a quello tridimensionale. Sebbene tecnicamente parlando entrambe le rappresentazioni siano in realtà tridimensionali, dato che il campo di visualizzazione è comunque uno schermo piano in entrambi i casi, la differenza principale tra queste due modalità è che la prima costringe gli utenti a pensare in termini di elementi geometrici piani e strutture piane (finestre, icone nelle finestre, bottoni disposti all'interno di una griglia bidimensionale), mentre la rappresentazione tridimensionale utilizza una proiezione prospettica per fornire un senso di spazialità ed immersione. Spesso questo paradigma utilizza un punto di vista soggettivo, vale a dire che l'utente comprende la propria posizione nello spazio vedendo una rappresentazione prospettica delle immediate vicinanze, ma senza la possibilità di vedere se stesso (esattamente come nella realtà).

La maggior parte dei pattern di collaborazione può essere realizzata sia in 2D che in 3D. Nel 3D, tuttavia, è possibile pensare un insieme di situazioni collaborative che sono impossibili da rappresentare in 2D, e che possono fornire una percezione aumentata dello spazio condiviso.

Il modello inizialmente proposto (definito "base") elenca i più importanti *elementi cooperativi* di base (gli elementi che insieme compongono una metafora cooperativa). Il modello "base" ha il vantaggio di essere relativamente semplice, ed è quello che è stato effettivamente usato come riferimento durante lo sviluppo e la ricerca di tutti gli altri moduli del framework collaborativo.

Tuttavia, durante il suo utilizzo, è emerso come nella presentazione di questo modello vi siano incogruenze, limitazioni, e incompletezze. Si è allora tentato di pensare un nuovo modello, detto "esteso", che tuttavia risulta di maggiore complessità, anche se più puntuale, e rimane al momento un modello di collaborazione puramente teorico, anche se un prossimo futuro passo potrebbe essere quello di tentarne l'implementazione.

Il modello "base" consiste essenzialmente nell'elencare una serie di elementi cooperativi e nello spiegare come possono essere combinati per formare delle metafore. Per alcuni di questi elementi, è possibile descrivere come potrebbero essere implementati in 2D o in 3D, e quali non hanno un corrispondente nella realtà (non

rappresentano cioè un modo di collaborare che può fisicamente avvenire nella realtà), ma possono essere utilizzati in modo efficace in particolari situazioni collaborative.

Gli elementi cooperativi afferiscono alle seguenti categorie:

- Distribuzione dello stato condiviso
- Meccanismo di raggruppamento
- Flusso Informativo
- Visualizzazione
- Movimento
- Autorappresentazione
- Comunicazione Testuale

### **3.2.1 Distribuzione dello stato condiviso**

Uno spazio condiviso è l'insieme delle informazioni circa lo stato di ciascun partecipante all'applicazione. Negli ambienti 3D, può essere la posizione di un avatar, il suo movimento, o il suo stato di quiete, oppure lo stato di una risorsa condivisa, come le scritte presenti su una whiteboard collaborativa. Ciascun utente può partecipare o essere isolato dallo stato condiviso (ed in questo caso non può far parte del meccanismo di collaborazione).

La condivisione dello stato può avvenire su due piani:

- *Condivisione Integrale*: Tutti gli utenti all'interno dell'applicazione condividono gli stessi eventi. Tutti hanno pertanto la stessa percezione dell'ambiente nel medesimo momento.
- *Condivisione Gruppale*: Lo stato condiviso può essere partizionato in sottogruppi; ciò equivale a dire che esistono più stati paralleli diversi per lo stesso ambiente. Ciascun partecipante può appartenere solamente ad una di queste partizioni di stato in un dato momento.

#### **Mutue Esclusioni**

Se si utilizza una condivisione integrale, non è possibile utilizzare un criterio di condivisione grupale, e viceversa.

#### **Miglioramenti di questo elemento cooperativo rispetto alla realtà**

La possibilità di creare applicazioni in cui per uno stesso ambiente esistono diversi possibili stati al medesimo tempo, non ha una controparte nella realtà. Si tratta di una possibilità interessante per creare situazioni multi-percorso in cui varie scelte possono dare come risultato eventi diversi.

### 3.2.2 Meccanismi di Raggruppamento

I meccanismi di raggruppamento si riferiscono a regole atte a gestire i gruppi di utenti. I gruppi sono un modo per tenere insieme gli utenti secondo i loro interessi o la loro attività; ciascun gruppo condivide particolari elementi cooperativi, e si comporta seguendo una metafora comune. Ciascun gruppo deve essere identificato da un nome univoco, pertanto, ad esempio, un gruppo che si incontra per discutere di arte moderna si potrà chiamare “ArteModerna”. La gestione dei gruppi avviene definendo quattro sottocategorie di proprietà:

- *Modalità di Creazione*: Regola la procedura di creazione di un gruppo.
  - o *Dinamica*: è possibile creare un gruppo non esistente in precedenza, senza alcuna restrizione. Il creatore è l'utente che ordina la creazione dinamica del gruppo.
  - o *Predeterminata*: il nome gruppo è predeterminato e già disponibile e non è possibile creare un nuovo gruppo con questo nome. Il creatore è l'operatore di sistema o un utente predeterminato che deve essere identificato mediante password (vedi *Gestione della Leadership*).
  
- *Modalità di Protezione*: Definisce accessi e privilegi in un gruppo
  - o *Libera*: è possibile accedere ad un gruppo senza restrizione
  - o *Password*: è possibile accedere al gruppo solo mediante password, decisa dal creatore.
  - o *Numero Chiuso*: solo un massimo numero di utenti possono partecipare simultaneamente allo stesso gruppo. Il numero chiuso è stabilito dal creatore.
  - o *Banning*: il creatore definisce una lista di utenti che non possono accedere al gruppo.
  - o *Kicking*: il creatore espelle forzatamente un utente dal gruppo.
  
- *Modalità di Distruzione*: Definisce i criteri con cui un gruppo deve essere distrutto.
  - o *Persistente*: Una volta creato, il gruppo non può mai essere distrutto, anche se vuoto.
  - o *Dinamico*: Il gruppo si distrugge quando non ospita più alcun utente
  - o *Forzato*: Il gruppo si distrugge e i partecipanti si disperdono su comando del creatore. Questo può accadere solo se c'è la disponibilità di almeno un gruppo libero, per accogliere tutti i partecipanti (solitamente denominato *gruppo globale*).
  
- *Gestione della Leadership*: in molte situazioni collaborative, gli utenti non collaborano allo stesso livello, ma uno o più partecipanti possono avere abilità speciali (mostrano la strada, indicano un oggetto, o sono gli unici partecipanti abilitati a parlare). La *leadership* modella un particolare privilegio all'interno della collaborazione, ed il meccanismo attraverso il quale la leadership viene acquisita e passata di mano.

- *Vincolata*: una volta che la leadership è acquisita, non può essere passata ad altri.
- *Protetta da Password*: la leadership può essere acquisita da tutti coloro che forniscono la password corretta.
- *Passaggio di Privilegio*: un utente che detiene la leadership può cederla ad un altro utente del gruppo.
- *Eredità*: utenti con leadership possono conferire leadership a uno o più altri utenti, senza cedere la leadership in loro possesso
- *Token*: la leadership è un token che può essere acquisito dal primo utente che la richiede. Ci sono tanti leader quanti sono i token di leadership disponibili. Perché qualcuno possa acquisire la leadership, ci deve essere un token disponibile. Gli utenti con leadership possono cederla, liberando un token.
- *Basata su creazione*: Nella modalità di formazione gruppo libera, la leadership viene automaticamente conferita all'utente che crea un nuovo gruppo libero.

### **Mutue Esclusioni**

Un gruppo può essere dinamico o predeterminato. Similmente, può essere libero oppure protetto da password, ma entrambi i tipi possono anche essere a numero chiuso, e dotati di banning e kicking. Solo un tipo di distruzione può essere specificato. La gestione della leadership può essere definita utilizzando più di un elemento alla volta; per esempio, la leadership può essere basata su creazione e vincolata, oppure basata su creazione e tokenizzata. Alcuni degli elementi di gestione non possono essere usati insieme, ad es. la leadership non può essere gestita contemporaneamente in modo Tokenizzato o Ereditato.

### **Miglioramenti di questo elemento cooperativo rispetto alla realtà**

La collaborazione di gruppo nella realtà è normalmente basata su creazione e con leadership vincolata, la creazione del gruppo è predeterminata e a numero chiuso, e la distruzione è dinamica. Nei gruppi reali, solitamente non si usano sistemi complessi di gestione della leadership. Nella collaborazione virtuale, invece, la leadership può essere distribuita all'interno del group e può essere utilizzata per controllare degli aspetti avanzati della collaborazione (ad es. per quanto riguarda la visualizzazione, la possibilità di cambiare i punti di vista dei soggetti).

### **3.2.3 Flusso Informativo**

Le regole di flusso informativo possono essere collegate a qualsiasi altro elemento cooperativo per specificare quali sono le direzioni e le modalità in cui l'informazione può essere trasmessa durante la collaborazione.

- *Pull*: l'informazione si trasmette dal fornitore di informazione al consumatore, perché il consumatore richiede esplicitamente al fornitore l'informazione stessa (*preleva* informazione dal fornitore).

- *Push*: l'informazione si trasmette dal fornitore al consumatore, perché il fornitore esplicitamente pubblica nuove informazioni verso il consumatore (*pubblicazione* dal fornitore al consumatore).

### **Mutue Esclusioni**

Le modalità di Push e di Pull possono essere utilizzate simultaneamente. Per esempio, un messaggio testuale o in genere un'informazione possono essere prelevate o pubblicate, anche se solitamente per i messaggi testuali avviene una pubblicazione verso l'utente quando nuovi messaggi sono disponibili (esattamente come le chiamate telefoniche sono pubblicate mediante uno squillo sul telefono cellulare). Le modalità di push e pull sono particolarmente efficaci dal punto di vista della visualizzazione, poiché consentono di decidere se un utente desidera vedere dal punto di vista ("attraverso gli occhi") di un altro utente, oppure se un utente desidera costringere gli altri utenti a vedere quello che egli vede (pubblicando la propria vista "negli occhi" degli altri). Inoltre, potrebbe essere possibile pubblicare la propria posizione agli altri, forzando in questo modo gli utenti a spostarsi in determinate posizioni dello spazio virtuale, e così via.

### **Miglioramenti di questo elemento cooperativo rispetto alla realtà**

Push e pull sono in generale modalità piuttosto comuni nella comunicazione vocale e testuale in realtà, ma una caratteristica unica degli ambienti virtuali è la loro applicabilità alla visualizzazione e alla posizione degli elementi.

### **3.2.4 Visualizzazione**

La modalità di visualizzazione indica le regole imposte per la gestione della visualizzazione della scena, sia come politica per l'intero mondo, che per il singolo utente.

- *Visualizzazione di scena*: si riferisce alle opzioni di visualizzazione degli agenti nella scena
  - o *Totale*: Visualizza tutti gli avatar presenti nella zona di interesse
  - o *Grupuale*: Visualizza solo gli avatar appartenenti al gruppo
  - o *Selezione*: Visualizza solo determinati avatar (a scelta)
  - o *Personale*: Visualizza il proprio avatar nella scena
- *Visualizzazione Soggettiva*: si riferisce al tipo di visualizzazione tridimensionale che si vuole utilizzare per la propria soggettiva.
  - o *Normale*: Visualizzazione soggettiva pura, 'dagli occhi'.
  - o *Prossima*: Visualizzazione da un punto esterno all'avatar, ma prossimo ad esso
  - o *Esterna*: La visualizzazione avviene da una telecamera esterna posizionata in un punto relativo alla posizione dell'avatar, che si muove dunque insieme ad esso.
  - o *Panoramica*: La visualizzazione avviene da una camera panoramica fissa, posizionata in un punto elevato dell'ambiente.

- *Avatar Esterno*: La visualizzazione avviene dagli occhi di un altro avatar a scelta (una guida oppure un amico).

### **Mutue Esclusioni**

Si possono usare tutte le visualizzazioni soggettive, oppure alcune di esse possono essere proibite per motivi legati alla natura dell'applicazione. Tuttavia, è possibile avere solo un tipo di visualizzazione alla volta. Le visualizzazioni di scena sono invece tutte mutuamente esclusive.

### **Miglioramenti di questo elemento cooperativo rispetto alla realtà**

Gli elementi di visualizzazione proposti costituiscono un deciso miglioramento rispetto all'esperienza quotidiana della collaborazione nella realtà. Nelle realtà virtuali è possibile vedersi agire, o incarnare altri utenti guardando attraverso i loro occhi. Questo aspetto ha enormi potenzialità dal punto di vista della realizzazione di giochi, dimostrazioni, e attività collaborative.

### **3.2.5 Movimento**

Si riferisce alle tipologie di movimento disponibili per l'utente. I movimenti possono essere limitati per scopi educativi o navigazionali all'interno di una particolare applicazione.

- *Avanzamento normale*: si tratta della normale modalità di spostamento, camminando, in corsa, o volando, per posizioni spaziali contigue.
- *Salto a posizione prossima*: salta accanto ad una posizione predeterminata, marcata da un avatar o da un oggetto.
- *Salto a zona*: salta all'interno di una zona predeterminata, marcata da un oggetto, un avatar, o un gruppo.

### **Mutue Esclusioni**

Tutte le modalità di movimento possono essere consentite contemporaneamente.

### **Miglioramenti di questo elemento cooperativo rispetto alla realtà**

Nella realtà l'elemento utilizzato è lo spostamento in intervalli continui di tempo e spazio. Nella virtualità esiste la possibilità degli ipersalti, che consente di avere meccanismi navigazionali più interessanti, anche se dannosi in termini collaborativi (un utente che esegue un ipersalto viene facilmente perso di vista dagli altri utenti).

### **3.2.6 Autorappresentazione**

L'autorappresentazione descrive il modo con cui il particolare stato dell'utente può essere rappresentato. Questi elementi possono essere abilitati o disabilitati in rapporto alla loro rilevanza all'interno del tipo di applicazione.

- *Emoticons*: gli emoticon sono brevi frasi o aggettivi, spesso accoppiati a simboli, disegni o animazioni, che rappresentano lo stato emozionale e i

sentimenti dell'utente. Sono utilizzati per rappresentare in modo più efficace un senso di un messaggio, migliorando in questo modo la comunicazione e la collaborazione. Tipici esempi sono \*smiles, \*hug, 8-), oppure le animazioni predefinite che si vedono in tanti ambienti virtuali giocosi (avatar che ballano la macarena, e via dicendo).

- *Grado di Presenza*: gli indicatori del grado di presenza indicano il coinvolgimento di un utente all'interno di un ambiente. La rappresentazione può essere testuale o simbolica. Alcuni esempi sono Busy, Do Not Disturb, Away, ecc.

### 3.2.7 Comunicazione Testuale

In molti ambienti virtuali, il sistema di comunicazione più efficiente è ancora testuale: gli utenti digitano quello che vogliono comunicare ad altri utenti, e leggono la relativa risposta. Alcuni ambienti offrono anche comunicazione vocale. Tuttavia, è difficoltoso navigare ed interagire in un ambiente sintetico 3D e contemporaneamente parlare e ascoltare, come dimostrato da numerose esperienze, soprattutto con un elevato numero di utenti. Per questo motivi, la maggior parte delle volte la comunicazione testuale è ancora il metodo preferito.

- *Libera*: un messaggio generato viene ridistribuito a tutti gli utenti presenti nell'ambiente virtuale condiviso, compreso il generatore. Vengono ricevuti tutti i messaggi inviati in modalità Libera.
- *Grupuale*: il messaggio generato viene ridistribuito a tutti i partecipanti allo stesso gruppo. Vengono ricevuti i messaggi inviati dagli altri partecipanti al medesimo gruppo
- *Grupuale Vincolato*: si ricevono i messaggi del creatore del gruppo, ma non è possibile instradare messaggi all'interno del gruppo da parte degli altri partecipanti.
- *Bisbiglio*: viene inviato un messaggio ad un destinatario specifico, indipendentemente dal gruppo di appartenenza. Il collegamento tra i due utenti è stabilito solo per la durata del messaggio, non è aperto alcun canale fisso.
- *Privato*: conversazione a due vie tra due partecipanti. Non sono ricevuti altri messaggi spediti in altre modalità.
- *Bisbiglio Allargato*: conversazione a ad una via diretta a più partecipanti, stabilita per la sola durata del messaggio. Le risposte arrivano dai destinatari mediante bisbiglio o instaurazione di un canale privato.

#### Mutue Esclusioni

La modalità libera, grupuale e grupuale vincolata sono in mutua esclusione. Il resto degli elementi possono essere usati liberamente in combinazione con questi primi tre.

#### Miglioramenti di questo elemento cooperativo rispetto alla realtà

Questi elementi modellano una conversazione così come può avvenire in un contesto reale di collaborazione; tuttavia, le modalità di bisbiglio allargato e gruppale vincolato possono essere più difficili da effettuare in realtà rispetto alla virtualità.

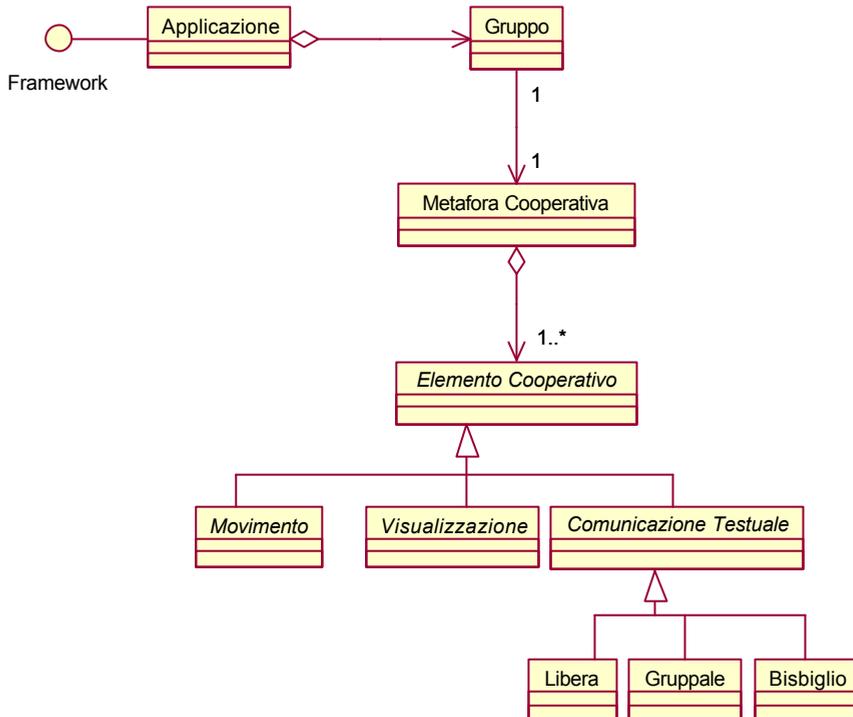


Fig 3-1: il diagramma del modello collaborativo nella versione base.

### 3.2.8 Esempio di Metafora Cooperativa

Una metafora cooperativa nel modello base si configura scegliendo un nome, spiegando quale è il suo scopo, ed enumerando quali sono gli elementi cooperativi che utilizza. A beneficio della comprensione di questo meccanismo, riportiamo un semplice esempio.

All'interno di una applicazione di visita museale tridimensionale, si può progettare una metafora cooperativa "Visita Guidata Programmata". Si tratta di un gruppo preformato, rigido, che accompagna altri utenti ad una visita ben specifica all'interno del mondo virtuale, con modalità che prevedono spiegazioni da parte della guida, ma non

consentono ai partecipanti di distrarsi parlando tra di loro. Sono previste solo brevi comunicazioni tra partecipanti del gruppo.

La metafora cooperativa si forma utilizzando questi elementi cooperativi:

*Condivisione - Integrale*

*Raggruppamento – Creazione Predeterminata, Numero Chiuso, Banning, Kicking*

*Visualizzazione – Scena Tutte; Soggettiva Tutte*

*Movimento – Avanzamento Normale, Salto Prossimo, Salto Zona*

*Com. Testuale – Gruppo Vincolato, Bisbiglio, Privato, Bisbiglio Allargato*

### **3.3 Il modello esteso**

Il pregio del modello base è la sua semplicità concettuale. L'idea di fondo è semplicemente quella di elencare caratteristiche tipiche della collaborazione negli ambienti virtuali, e quali vantaggi o caratteristiche aggiuntive dispongono rispetto al tipo di collaborazione in ambiente reale che vogliono modellare.

La semplicità concettuale corrisponde alla possibilità di implementarle tecnicamente più agevolmente, sia nella loro realizzazione, sia nel disegno di un formalismo che descriva al sistema quali modalità attivare e quali no (si veda al successivo capitolo per il sistema di configurazione). Tuttavia essa è anche causa di alcune limitatezze, che, almeno concettualmente, sembrano diminuirne la validità. Si è allora cercato di riformulare il modello estendendolo, anche se non si è ancora definito un linguaggio formale di descrizione valido che possa essere generato da un opportuno configuratore. I tentativi finora fatti in tal proposito, hanno generato linguaggi di complessità eccessiva, che ne pregiudicano l'effettiva usabilità.

In particolare, il modello esteso corregge quello base nei seguenti modi:

- introduce una terminologia maggiormente precisa per definire gli aspetti della collaborazione, introducendo il concetto di classe
- ridefinisce gli elementi cooperativi organizzandoli per classi
- prevede che una metafora di collaborazione non sia staticamente agganciata ad un gruppo di persone, ma sia anche legata all'ambiente in cui esse si trovano, e ad una determinata situazione. La metafora di collaborazione del gruppo può cambiare in relazione agli eventi
- individua due fasi di progettazione della metafora cooperativa, la configurazione e l'istanziamento, e definisce il concetto di scenario, considerando la definizione delle regole di gestione del gruppo non come un elemento della cooperazione ma come una modalità di implementazione della metafora cooperativa.

#### **3.3.1 Definizioni**

Vengono fornite di seguito alcune definizioni relative al modello esteso.

Un *framework* o *sistema* è l'insieme degli apparati concettuali e tecnici atti a concepire, realizzare ed utilizzare un sistema di collaborazione condiviso con rappresentazione tridimensionale. Esso può comprendere sia modelli concettuali (come le *metafore visive* o le *metafore cooperative*), sia componenti software organizzati secondo un determinato progetto.

Una *applicazione* è un insieme di contenuti informativi, di modelli tridimensionali rappresentanti ambienti e oggetti, e una politica di fruizione delle informazioni e del modello, atta a consentire agli utenti lo svolgimento di una determinata attività. Ad esempio, la creazione di un museo virtuale tridimensionale collaborativo dedicato a Leonardo da Vinci.

Un *componente dell'applicazione* è un elemento strutturale che compone l'applicazione. Esso può essere un *utente*, un *contesto*, una *entità*.

Un *utente* (o avatar) rappresenta un singolo utilizzatore del sistema che si muove, interagisce e collabora all'interno dell'applicazione.

Un *contesto* è una porzione di mondo con determinate proprietà. Ogni porzione di mondo è associata ad uno ed un solo contesto.

Un'entità è la rappresentazione grafica di un singolo oggetto del mondo (che non sia un avatar).

Una *classe* è una generale categoria di componenti dell'applicazione. Le classi si dividono in tre sottocategorie:

- *classe di utenti*: descrive una particolare categoria di utilizzatori del sistema. Viene anche definito *ruolo* dell'utente nella collaborazione.
- *classe di contesto*: descrive una particolare categoria di ambienti realizzati nello spazio di navigazione
- *classe di entità*: descrive particolari categorie di entità presenti nello spazio di navigazione.

Utenti, contesti o entità appartenenti ad una determinata classe hanno uguali proprietà.

La progettazione della cooperazione in un mondo virtuale si compone di due fasi:

- la *fase di configurazione* comprende l'analisi dei requisiti che la cooperazione deve soddisfare, l'identificazione delle classi di utente, di contesto e di entità, la progettazione di tutte le metafore cooperative presenti nello scenario e la definizione dei gruppi.
- la *fase di istanziazione* consiste nel definire le regole di passaggio da una metafora all'altra all'interno dello stesso scenario, e nel determinare le politiche di creazione, protezione e distruzione dei gruppi, con assegnamento degli account (nome utente e password per l'accesso all'applicazione) qualora fosse necessario.

Nella fase di configurazione, il passo iniziale è dato dall'*analisi dei requisiti* dell'applicazione che si vuole realizzare. La natura dell'applicazione potrebbe essere varia: potrebbe essere un museo che si vuole rendere visitabile via Internet, così da

realizzare delle visite virtuali all'interno della ricostruzione tridimensionale dello stesso, oppure un negozio che vuole aprire il suo sito di commercio elettronico (sempre sotto forma di mondo virtuale, non un tradizionale sito bidimensionale), o un centro di addestramento di gruppi che devono imparare ad operare in modo sincronizzato su macchinari complessi, o orientarsi all'interno di strutture complesse.

L'analisi deve individuare le classi di utente (ruoli), le classi di contesto, le classi di entità, e le metafore cooperative che si vogliono realizzare.

Un *ruolo* può essere generalmente definito come l'insieme delle operazioni che un utente può compiere e di quelle che gli sono interdette. L'assegnamento dei ruoli agli utenti è fondamentale perché si realizzi la cooperazione desiderata: ciascun utente, infatti, collabora con gli altri comportandosi in relazione al ruolo che deve rivestire all'interno dell'applicazione (docente, studente, istruttore, ecc.).

Il concetto di ruolo identifica la classe di utenti: tutti gli utenti appartenenti ad una classe sono soggetti alle medesime regole di collaborazione con altri utenti e con altre entità, in relazione al contesto in cui si trovano in un dato istante. Nell'esempio del museo virtuale, il ruolo di "guida museale" consiste nell'accompagnare i visitatori all'interno del mondo virtuale, avendo a disposizione delle particolari modalità di collaborazione atte a simulare questa attività.

Successivamente si determinano le *classi di contesto* in cui si divide l'ambiente da rappresentare. Ciascuna classe di contesto può essere messa in corrispondenza con un *pattern spaziale* identificato seguendo il modello della *metafora visiva*. Il contesto consente di differenziare l'interazione tra due ruoli distinti, in relazione dell'ambiente in cui si trovano ad interagire. Ad esempio l'interazione in un corridoio di ingresso, in cui tutti possono parlare tra loro, sia docenti, che studenti, cambia quando si entra nella sala della conferenza, dove i docenti parlano agli studenti, ma gli studenti non possono parlare ai docenti se non previo un esplicito meccanismo di prenotazione di intervento.

Vanno poi assegnate delle *classi di entità* ai principali oggetti componenti il mondo, con i quali si prevede che gli utenti possano avere delle interazioni. La suddivisione in categorie consente di modellare in modo diverso le modalità con cui la collaborazione con questi oggetti debba avvenire, o l'effetto che lo stato di questi oggetti deve avere su altri oggetti o utenti.

Una *metafora cooperativa*, nella visione del modello esteso, è l'insieme delle modalità di interazione tra le classi di utente, di contesto e di entità, definite attraverso gli elementi cooperativi, con lo scopo di regolamentare il comportamento dei vari componenti applicativi del mondo favorendo e stimolando uno o più tipi di cooperazione. Un *elemento cooperativo* è una regola base di interazione tra elementi del mondo.

Gli elementi cooperativi possono descrivere diversi tipi di interazioni, fondamentalmente tra utente ed utente, tra utente e contesto, tra utente ed entità e tra

entità ed entità. Un elemento cooperativo è un'entità atomica, e non è scomponibile in ulteriori parti, ma rappresenta la minima funzione implementabile e pensabile per comporre un modalità di collaborazione complessa.

Alcuni esempi di elementi di cooperazione tra classe di utente e classe di utente sono i seguenti:

- *Messaging*: possibilità di mandare un messaggio di testo da un utente ad un altro
- *Approaching*: possibilità per un utente di avvicinarsi automaticamente ad un altro utente nel suo campo di visualizzazione
- *Emoting*: possibilità per un utente di comunicare con un altro utente attraverso l'uso di emoticons, ovvero di espressioni facciali o simboli che esprimono uno stato d'animo
- *Speaking*: possibilità per un utente di comunicare con un altro utente attraverso un canale vocale
- *Gesturing*: possibilità per un utente di comunicare con un altro utente attraverso l'uso di gesticolazioni corporali più complesse degli usuali emoticons
- *Kicking*: possibilità per un utente di espellere dal mondo un altro utente
- *Banning*: possibilità per un utente di bandire un altro utente proibendogli l'accesso al mondo in questione
- *Calling*: possibilità per un utente di chiamare a sé un altro utente, spostandone fisicamente la posizione in sua prossimità
- *Viewing*: possibilità per un utente di vedere un altro utente presente nel mondo nel momento in cui passa per il suo campo visivo
- *Embodying*: possibilità per un utente di vedere con gli occhi di un altro utente, facendo coincidere il proprio punto di vista con quello dell'utente prescelto
- *Presence Showing*: possibilità di segnalare ad un altro utente il proprio grado di presenza, ossia la propria disponibilità a parlare o a cooperare, oppure l'indisponibilità per assenza o per altra ragione
- *Tethering*: possibilità per un utente di "legarsi" ad un altro utente, ossia di essere guidato in modo passivo all'esplorazione del mondo

Alcuni esempi di elementi di cooperazione tra classe di utente e classe di entità sono i seguenti:

- *Approaching*: possibilità per un utente di avvicinarsi automaticamente ad un'entità nel suo campo di visualizzazione, venendone eventualmente "attratto"
- *Taking*: possibilità per un utente di prendere con sé un'entità del mondo, e ovviamente anche di riporla nel mondo
- *Moving*: possibilità per un utente di cambiare la posizione nel mondo di un'entità
- *Collision Enabling*: possibilità per un utente di collidere o meno con un'entità, nel caso di non abilitazione l'utente può passare attraverso l'entità

- *Messaging*: possibilità per un utente di mandare un messaggio di testo ad una entità; questo elemento può risultare utile qualora si utilizzassero delle entità in grado di interagire con l'utente come per esempio degli agenti automatici
- *Interaction*: questo elemento consente una interazione mediante un'azione. La natura dell'interazione varia in relazione al behavior o comportamento associato all'entità (v. oltre per il concetto di behavior).
- *Tethering*: possibilità per un utente di "legarsi" ad un'entità, facendosi trasportare da essa

Alcuni esempi di elementi di cooperazione tra classe di entità e classe di entità sono i seguenti:

- *Collision enabling*: possibilità per un'entità di collidere o meno con un'altra entità, nel caso di non abilitazione le due entità possono compenetrarsi; questo può essere utile in un eventuale utilizzo nell'applicazione di agenti automatici
- *Interaction*: questo elemento consente una non meglio specificata interazione complessa tra un'entità ed un'altra entità, la sua utilità va sempre collegata ad un eventuale utilizzo nell'applicazione di agenti
- *Tethering*: possibilità per un'entità di "legarsi" ad un'altra entità, facendosi trasportare da essa

Alcuni esempi di elementi di cooperazione tra classe di utente e classe di contesto sono i seguenti:

- *Camera viewing*: possibilità da parte dell'utente di posizionare delle "camere", punti di vista fissi, per visualizzare l'ambiente da determinati punti di vista
- *Moving*: possibilità da parte dell'utente di muoversi spazialmente all'interno del contesto
- *Door access*: possibilità di entrare nel contesto attraverso un ingresso
- *Door leaving*: possibilità di uscire dal contesto attraverso un'uscita
- *Teleaccessing*: possibilità di entrare nel contesto mediante un ipersalto
- *Teleleaving*: possibilità di uscire dal contesto mediante un ipersalto
- *Gravity*: attivazione o meno della gravità all'interno del contesto. Senza la gravità, si lascia libero l'utente di esplorare lo spazio senza essere vincolato a seguire la modellazione del terreno offerta per il contesto.

Definite le classi di utente, contesto, ed entità, bisogna assegnare le loro proprietà collaborative specificando quali tra questi elementi cooperativi sono attivi o pertinenti ad esse. L'assegnamento avviene specificando come interagisce ciascuna classe con l'altra, ad esempio quali elementi cooperativi sono attivi quando una data classe di utente interagisce con un'altra classe di utente, o quali quando una classe di utente interagisce con una classe di contesto, e così via.

Successivamente occorre definire come devono essere composti i gruppi di utenti che possono partecipare all'applicazione. Un gruppo è definito come un insieme di utenti che partecipano al medesimo scenario cooperativo, condividendone lo spazio. La

definizione dei gruppi consiste nel decidere quanti gruppi possono visitare l'applicazione, qual è il loro nome, quali sono le loro caratteristiche in termini di scenario collaborativo ad esso assegnato. Viene inoltre definito qual è il numero minimo e massimo di partecipanti appartenenti a un dato ruolo, che possono essere ammessi in un dato gruppo (cardinalità per classe utente del gruppo). Inoltre viene definito se per essere ammessi in un determinato ruolo occorra una identificazione. Qualora fosse richiesta l'identificazione per la partecipazione di un utente entro un dato ruolo, in fase di istanziazione dovrebbero essere assegnate ad ogni utente un nome ed una password. Questa fase consente di stabilire il tipo di utenti che possono accedere all'applicazione, ed eventualmente stabilire esattamente le loro identità, per assegnare il ruolo che compete loro all'interno dell'applicazione (insegnante, guida, studente, istruttore).

Una delle aggiunte del modello esteso, e il motivo principale per cui il modello base è stato ripensato, è il concetto che per un dato ruolo possono essere applicate più metafore cooperative, in relazione al contesto. Pertanto il concetto di metafora cooperativa non è più legato al gruppo di appartenenza, come nel modello concettuale "base", ma piuttosto alla contingenza della situazione collaborativa, modellando quindi ancora più fedelmente i meccanismi di collaborazione che avvengono nella realtà.

La possibilità di avere più metafore cooperative costringe a riprogettare l'impiantito del modello collaborativo: la metafora cooperativa non è più il concetto "contenitore" gerarchicamente più alto, ma è solo una delle possibili configurazioni di interazione, in un determinato contesto ed in un determinato momento.

La raccolta delle metafore cooperative possibili, più di altri elementi necessari al governo della collaborazione, viene detto *scenario cooperativo*.

La definizione dello scenario cooperativo avviene nel contesto della seconda fase di definizione della collaborazione, la fase di *istanziazione*.

Definiamo *scenario cooperativo* l'insieme dei seguenti aspetti:

- la definizione di tutte le metafore cooperative che vi appartengono, avendo in ciascuna definito quali elementi cooperativi sono attivi per quali classi di entità, classi di contesto e classi di utenti esistenti
- la definizione delle regole che innescano il cambio da una metafora cooperativa all'altra, poiché non è possibile che più di una metafora cooperativa sia valida nel medesimo tempo, pena il verificarsi di situazioni contraddittorie
- la definizione delle regole di gestione dei gruppi valide per i gruppi identificati e censiti nelle fasi precedenti. Queste regole di gestione seguono le regole del modello collaborativo "base".

Il passaggio da una metafora all'altra può avvenire in tre modi:

- *a comando*: attraverso un comando espresso da un certo utente, che ricopre un ruolo in cui è possibile emettere questo tipo di comando

- *a tempo*: attraverso un *trigger* temporale – dopo una certa quantità di tempo in una data modalità collaborativa, si passa ad una successiva
- *a posizione*: attraverso un *trigger* posizionale montato su una entità – passando una determinata soglia o entrando in un determinato contesto, si cambia la metafora di cooperazione.

Nella fase di istanziiazione, dunque, si esplicitano quali delle metafore cooperative entrano all'interno di uno scenario, e si specificano le regole di cambio delle metafore.

Il secondo aspetto da definire nello scenario durante la fase di istanziiazione, è stabilire quali delle politiche di gestione del gruppo (definite come nel modello “base”) devono essere assegnate ai generali gruppi individuati in fase di configurazione. Contestualmente, se esiste l'esigenza di restringere l'accesso a particolari utenti identificati, che devono assumere determinati ruoli, in questa fase si devono elencare quali utenti (in termini di nome e password) devono afferire a quali gruppi.

Al termine della fase di istanziiazione, lo scenario, inteso come contenitore di tutte le definizioni ed i concetti sopra esposti, descrive in modo molto realistico come dovrebbe svolgersi la collaborazione all'interno dell'applicazione.

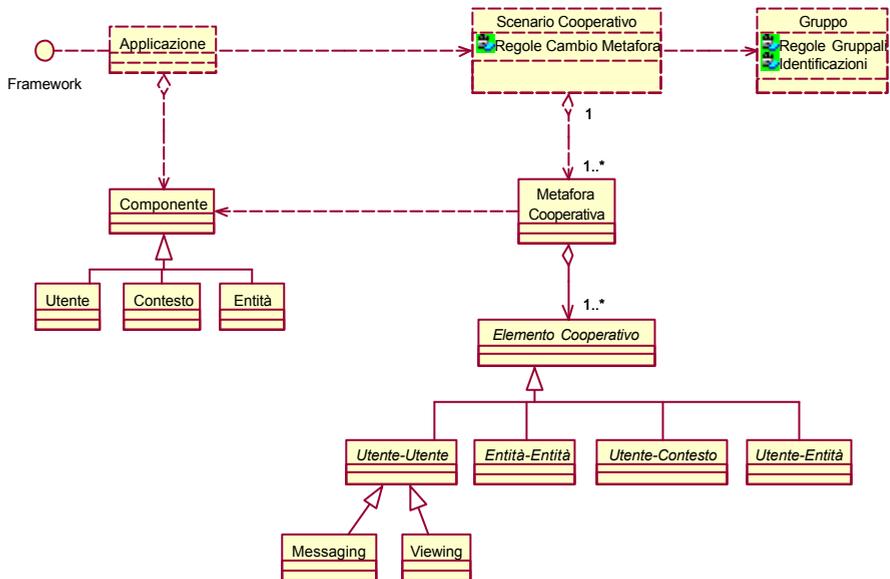


Fig 3-2: diagramma del modello collaborativo nella versione estesa.

### 3.3.2 Esempio applicativo

Consideriamo nuovamente un esempio di ambiente virtuale museale, e una modellazione di una collaborazione in cui un gruppo di utenti accede all'applicazione all'interno di una visita virtuale guidata da un esperto.

Innanzitutto, vanno individuate le classi di utente, contesto ed entità necessarie.

Classi di Utente:

Visitatore

Guida

Classi di Contesto:

Esposizione

Classi di Entità:

Oggetto Esposto

Oggetto Interattivo

Fanno parte della classe "Visitatore" gli utenti comuni: nel caso di una scolaresca saranno studenti, nel caso di una comitiva saranno le persone partecipanti alla visita che non siano né la guida né il loro capogruppo. Il ruolo "Guida" è evidentemente adibito alle guide museali, quindi ad utenti in grado di condurre gli altri visitatori in tour per il museo.

La classe di contesto "Esposizione" va riferita a tutti quegli ambienti virtuali adibiti a sala di esposizione: si può immaginare che ogni sala di esposizione del museo sia una stanza nel mondo virtuale.

Due sono le classi di entità individuate: la prima, "Oggetto Esposto", è riferita a tutti gli oggetti in esposizione nel museo; la seconda, "Oggetto Interattivo", è riferita invece ad eventuali oggetti manipolabili dall'utente in modo interattivo per ottenere nuove informazioni (ad esempio una macchina virtuale di Leonardo funzionante).

Il secondo passo della fase di configurazione consiste nel determinare quali sono le metafore cooperative che si vogliono creare. Ad esempio, si può definire la metafora "Visita Guidata", adatta per consentire agli utenti la visita al museo con l'ausilio della guida; desideriamo che questa visita sia da parte di classi di studenti, e sia regolamentata in modo che sia necessario seguire la guida, e non sia possibile abbandonare il gruppo per esplorare in solitario il mondo. Una seconda metafora, "Visita Libera", può consentire invece di esplorare il mondo senza essere legati a altri particolari utenti.

Lo scopo di queste due metafore è rappresentare una situazione di collaborazione in cui prima si effettua una visita guidata, e al termine delle spiegazioni diventa possibile

continuare la visita in modo autonomo. Per descrivere il funzionamento delle due metafore, vi si assegnano gli elementi cooperativi, spiegando come sono attivati negli incroci tra ciascuna delle classi dei tre componenti della collaborazione.

**Definizione Metafora “Visita Guidata”**

Classe Utente con Classe Utente:

Tra Visitatore e Visitatore:

Viewing

Tra Guida e Visitatore:

Messaging, Viewing, Embodying, Tethering

Tra Guida e Guida:

Messaging, Approaching, Calling, Viewing, Embodying, Tethering

Classe Utente con Classe Entità:

Tra Visitatore e Oggetto Esposto:

Approaching, Collision

Tra Guida e Oggetto Esposto:

Approaching, Collision

Tra Visitatore e Oggetto Interattivo:

Approaching, Collision

Tra Guida e Oggetto Interattivo:

Approaching, Collision, Interaction, Taking, Moving

Classe Entità con Classe Entità:

Non Rilevante

Classe Utente con Classe Contesto:

Tra Visitatore e Contesto Espositivo:

Camera, Moving, Door Access, Door Leaving, Gravity

Tra Guida e Contesto Espositivo:

Camera, Moving, Door Access, Door Leaving, TeleAccessing, Teleleaving

La visita guidata è concepita per dare massima libertà alla guida, che ha compito di spiegare i contenuti, tenendo vincolati i visitatori, che nella fase guidata devono solamente osservare quanto indicato ed ascoltare le spiegazioni. Per questo motivo, tra visitatore e visitatore non è consentito parlare: è possibile parlare solamente con la guida per rivolgere delle domande. Per il visitatore è però possibile cambiare il proprio punto di vista ed entrare nella visualizzazione della guida per osservare quello che essa sta osservando. Alla guida è concesso prendere e spostare gli oggetti, e interagire con essi durante la spiegazione: azioni non concesse ai visitatori, che vi si possono solamente avvicinare.

**Definizione Metafora “Visita Libera”**

Classe Utente con Classe Utente:

Tra Visitatore e Visitatore:

Messaging, Viewing, Embodying, Tethering  
 Tra Guida e Visitatore:  
 Messaging, Viewing, Embodying, Tethering  
 Tra Guida e Guida:  
 Messaging, Approaching, Calling, Viewing, Embodying, Tethering

Classe Utente con Classe Entità:  
 Tra Visitatore e Oggetto Esposto:  
 Approaching, Collision  
 Tra Guida e Oggetto Esposto:  
 Approaching, Collision  
 Tra Visitatore e Oggetto Interattivo:  
 Approaching, Collision, Interaction  
 Tra Guida e Oggetto Interattivo:  
 Approaching, Collision, Interaction, Taking, Moving

Classe Entità con Classe Entità:  
 Non Rilevante

Classe Utente con Classe Contesto:  
 Tra Visitatore e Contesto Espositivo:  
 Camera, Moving, Door Access, Door Leaving, Teleaccessing,  
 Teleleaving  
 Tra Guida e Contesto Espositivo:  
 Camera, Moving, Door Access, Door Leaving, TeleAccessing,  
 Teleleaving

La metafora di “Visita libera” conferisce al visitatore maggiori facoltà: consente ai visitatori di parlare tra loro, e di interagire con gli oggetti (ma non di prenderli o spostarli). Inoltre possono utilizzare le funzionalità di ipersalto e navigare senza sottostare ai vincoli della gravità, esplorando completamente lo spazio.

Definite le metafore cooperative, occorre ora definire le caratteristiche dei gruppi e le loro cardinalità.

Ad esempio, il gruppo “Visite Museo” può contenere una e una sola guida, per cui è necessaria una identificazione. Inoltre può contenere da uno a massimo quindici visitatori. Per i visitatori non è necessaria l’identificazione.

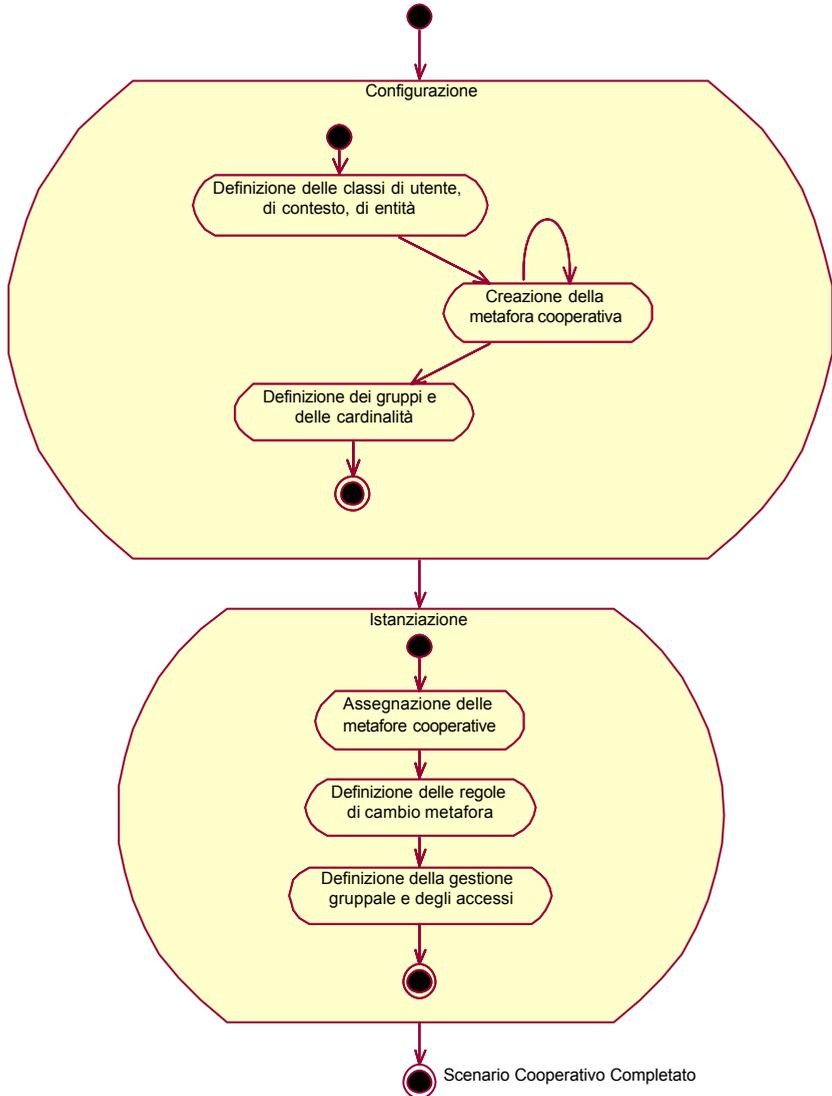
L’ultimo passo riguarda la definizione dello scenario cooperativo. Appartengono allo scenario cooperativo le metafore cooperative “Visita Guidata” e “Visita Libera”. Possono utilizzare queste metafore i partecipanti al gruppo “Visite Museo”. Il cambio tra Visita Guida e Visita Libera deve avvenire su comando della guida. Quando la guida termina la spiegazione, può innescare il passaggio alla visita libera per tutti i partecipanti del gruppo. Il ruolo di guida possono essere assunti dal dott. Mario Rossi,

con username “mrossi” e password “red”, e dal dott. Giuseppe Bianchi, con username “gbianchi” e password “white”.

Con la definizione dello scenario, si completa la descrizione di come deve avvenire la collaborazione in una determinata applicazione, utilizzando come linea guida il modello esteso delle metafore cooperative.

Nell’ambito della nostra ricerca sono stati realizzati dei programmi che guidano nella scelta di questi aspetti e generano delle strutture XML che descrivono queste scelte. Tuttavia, la struttura XML generata risulta ancora troppo complessa e non adatta ad una semplice implementazione da parte di un generico framework di supporto ai sistemi collaborativi tridimensionali.

Sebbene il modello “esteso” sia sicuramente interessante e più completo, il framework attuale implementa collaborazioni utilizzando parte del modello “base”. Una continuazione della ricerca porterà a tentativi di implementare il modello “esteso” sul framework collaborativo generale.



**Fig 3-3: le attività necessarie per descrivere una collaborazione utilizzando il modello collaborativo esteso.**

## 4 La gestione dei contenuti ed il Web Collaborativo

### 4.1 Premessa

Una delle applicazioni più interessanti dei sistemi collaborativi tridimensionali è la possibilità di accedere a contenuti informativi presenti su siti web tradizionali mediante un procedimento collaborativo, sfruttando cioè l'interazione con altre persone. Il metodo tradizionale di navigazione imposto dallo standard WWW consiste ad oggi nella fruizione delle informazioni sotto forma di pagine web in modo solitario da parte di ciascun utente. L'informazione viene esplorata seguendo una catena di collegamenti ipertestuali, e ricercata utilizzando opportuni motori di ricerca utilizzando delle parole chiavi per l'individuazione dell'informazione voluta.

L'utilizzo di un sistema di rappresentazione tridimensionale collaborativa per ricercare e esplorare le informazioni potrebbe essere tuttavia di interesse in casi in cui non si ha esattamente nozione di che cosa si sta cercando, oppure la natura dell'informazione ha un grado di complessità tale che il suo reperimento o la scelta di quale informazione utilizzare potrebbe avvenire in modo più efficace sotto la guida o la consulenza di altre persone più esperte.

Nei casi dunque in cui la natura dell'informazione è particolarmente grafica e strutturata, e potrebbe proficuamente avvalersi di rappresentazioni tridimensionali, l'uso a questo scopo di sistemi CVE rappresenta un'opportunità da considerare. Per utilizzare uno scenario di esempio, un utente potrebbe visitare una rappresentazione tridimensionale di un museo, accompagnato da altri visitatori e da una guida, in grado di indicare gli oggetti più interessanti e spiegarne la natura, o di rispondere direttamente ad alcune domande. Nel contesto tridimensionale, è possibile ricercare l'informazione navigando per stanze anziché utilizzando chiavi di ricerca. Una volta trovata una stanza o un oggetto di interesse, è possibile aprire la relativa informazione bidimensionale per ottenere maggiori informazioni. In linea generale, la tradizionale pagina contenente testo, immagini e video è naturalmente la forma migliore di presentazione, quando si tratta di trasmettere una grande quantità di informazione. L'approccio tridimensionale, invece, risulta ottimale quando si tratta di fornire rappresentazioni grafiche, strutturate, e di modellare situazioni di interazione. Rimane tuttavia la questione di mappare la struttura del sito web e dell'informazione in esso contenuta, con i relativi meccanismi di navigazione collegati, con la rappresentazione di una struttura di accesso tridimensionale alle medesime informazioni. In altre parole, individuare quale sia il metodo migliore per creare uno strato di accesso collaborativo tridimensionale su un contenuto progettato per essere letto, ed utilizzato, in modo bidimensionale.

## 4.2 La modellazione logica e le strutture di accesso

Il modello HDM<sup>13</sup> (Hypermedia Design Model) è un semiformalismo che consente di progettare l'organizzazione e la natura dei contenuti, solitamente (ma non solo) accessibili attraverso un sito web bidimensionale. L'utilità dell'adozione di un formalismo sta nel poter progettare a priori l'organizzazione dei contenuti, rilevandone le caratteristiche comuni, e decidendo in che modo deve essere possibile fruirne, progettando la navigazione ed i collegamenti tra i vari elementi costituenti i contenuti.

HDM affronta il problema della progettazione di contenuti fruibili in modo ipermediale (vale a dire, utilizzando le tecnologie e le metodologie tipiche dell'attuale multimedialità, combinate con le possibilità di incrocio e di collegamento dei contenuti tipici degli ipertesti) da diversi punti di vista:

- la struttura di ciascun elemento di contenuto
- le relazioni che intercorrono tra di essi
- i meccanismi di navigazione esistenti per la loro consultazione

In particolare la fase di progettazione si divide fondamentalmente in due fasi: la progettazione *in the large* e quella *in the small*. Nella progettazione in grande si descrive la struttura generale dell'applicazione, in termini di accesso, contenuti generali, e modalità di navigazione. Nella progettazione in piccolo si identificano invece con maggior precisione come organizzare ciascuna tipologia di contenuti, quali risorse multimediali raccogliere per ciascuna di esse, e come interconnetterle tra loro.

Questa metodologia fa dunque una distinzione molto importante: da una parte prevede la presenza di una *hyperbase*, vale a dire una raccolta di oggetti di informazioni, ospitanti il contenuto che costituisce l'oggetto ultimo dell'interazione e della navigazione, e dall'altra si appoggia su una *access base*, vale a dire le strutture progettate per navigare ed interconnettere tra loro le informazioni.

In particolare, i contenuti sono suddivisi in una struttura gerarchica così composta:

- *Entity*: un'unità informativa logicamente percepibile dall'utente, atta a fornire un'informazione compiuta.
- *Component*: componente di una entity, un insieme di più component consente di completare una entity, ma nessuna di esse ha senso compiuto singolarmente, al di fuori della entity.
- *Node*: E' l'effettivo "contenuto" inteso come risorsa multimediale, e rappresenta l'informazione finale ed atomica che viene presentata all'utente alla fine della navigazione attraverso la struttura. Essi possono essere interconnessi tra loro mediante la specifica di opportuni *link strutturali*.

Le Entity ed i Component possono avere dei tipi generali di appartenenza a cui fare riferimento, consentendo di creare dei template di presentazione dei contenuti.

---

<sup>13</sup> Garzotto F., Paolini P., Schwabe D. - HDM - A Model Based Approach to Hypermedia Application Design - ACM Transactions on Information Systems, Vol.11, N.1, Jan. 1993]

Il concetto di *Collection*, appartenente alle Strutture di Accesso, consente di rappresentare una raccolta di oggetti (membri della collezione), che possono essere entità, componenti, nodi o altre collezioni. Una collezione crea un “tema navigazionale” per l’utente, riferendosi ad un concetto comune secondo il quale gli oggetti sono stati raggruppati (ad esempio, la collezione “Cultura del Seicento” può comprendere brani musicali, testi, pitture del seicento, e ciascuna entità “brano”, “testo”, “pittura”, può a sua volta essere scomposta in una particolare struttura di componenti e nodi che normano come deve essere presentato e strutturato il contenuto per la sua fruizione). Una collezione può avere un Centro di Collezione, il quale rappresenta un particolare componente che funge da punto concentratore di accesso e di ritorno durante la navigazione della collezione stessa.

Elementi della progettazione *in piccolo* sono invece i seguenti:

- *Slot*: E’ un’informazione atomica che costituisce un elemento a diretto contatto con l’utente, può esistere in diverse forme, ed è quindi necessario definirne il tipo a priori; ci possono essere, infatti, slot semplici, caratterizzati da tipi base (es.: interi o caratteri) o complessi (es.: video o audio). Anche la molteplicità può variare e vi sono casi di slot singoli o multipli. E’ ammessa anche l’aggregazione di più slot associati.
- *Data Segment*: E’ un particolare insieme di slot associato ad un componente. Esistono diversi tipi base, predefiniti, di *Data Segment*: *Base Segment*, *Visual Segment*, *List Segment*, *Query Segment*. Anche in questo caso esiste il concetto di *Slot Type* e quello di *Data Segment Type* per raggruppare le istanze dei rispettivi *Slot* e *Data Segment* con proprietà comuni.

La progettazione dei contenuti e della struttura ipermediale utilizzando la metodologia HDM può avvenire mediante l’uso di un editor<sup>14</sup>, il quale consente di generare una descrizione testuale (utilizzando una struttura XML) del modello HDM, ad uso di un generatore automatico di siti web, la piattaforma JWEB<sup>15</sup>.

Generando il sito a partire dai contenuti memorizzati nell’hyperbase, ed organizzati secondo la struttura e le modalità navigazionali descritte con il modello HDM, è possibile generare rapidamente siti web bidimensionali (o, in generale, applicazioni ipermediali) con un elevato grado di consistenza e con una ben precisa struttura logica, frutto di una progettazione a priori guidata dalla metodologia.

### **4.3 La Navigazione 3D sui contenuti**

La disponibilità di una struttura di accesso e di contenuti già progettata risulta preziosa anche per la creazione di uno strato di accesso tridimensionale, che consenta di arrivare

---

<sup>14</sup> “Cap. 13: Tools for Designer”. In *Hypertext/Hypermedia Handbook*, Berk, E. and Devlin, J. (Eds.), McGraw Hill, 1991 *Garzotto, F., Paolini P., Schwabe D., and Berstein M.*

<sup>15</sup> Mario A. Bochicchio, Roberto Paiano, Paolo Paolini: *JWeb: An HDM Environment for Fast Development of Web Applications*. ICMCS, Vol. 2 1999: 809-813

ai contenuti 2D attraverso un'attività di collaborazione. Una collaborazione naturalmente arricchita e potenziata da tutti i meccanismi di percezione della condivisione che risultano aumentati, come detto, all'interno di un ambiente virtuale semi-immersivo. E' dunque pensabile ricercare una corrispondenza tra i generici concetti strutturali e navigazionali individuati da HDM, e, così come la generazione dinamica di siti web operata da Jweb opera una traduzione su un contesto bidimensionale, ottenere una generazione di una struttura di accesso tridimensionale, traducendo in elementi visivi spaziali le strutture di accesso concettuali previste dai progettisti dei contenuti.

Non sempre, tuttavia, la natura dei contenuti è tale per cui un'operazione di questo genere possa risultare vantaggiosa. In generale, la fruizione dell'informazione effettiva risulta più efficace nelle due dimensioni, esattamente come nella realtà siamo abituati a leggere un testo o a visionare un filmato. Tuttavia, in tutti quei casi in cui la ricerca dell'informazione risulta difficile semplicemente mediante la consultazione di un elenco categorizzato, oppure attraverso l'inserimento di parole chiave, lo strato di accesso collaborativo può risultare prezioso.

Sono questi i casi in cui i contenuti proposti – ad esempio - sono altamente specializzati, e richiedono una conoscenza della materia per poter trovare la giusta informazione. In questo caso si può ipotizzare l'instaurarsi di una collaborazione tra un assistente presente nel sito ed uno o più utenti in cerca dell'informazione: l'assistente, mediante domande ed interagendo con gli utenti, può proporre o condurre all'informazione desiderata.

E' questo un paradigma attuabile in due dimensioni utilizzando strumenti di assistenza web quali chat in linea, ICQ Surf, Net2Gether, LivePerson<sup>16</sup>, e via dicendo; tuttavia, mediante le tre dimensioni, è possibile aggiungere un maggiore livello di comunicazione, ed eventualmente utilizzare le possibilità di visualizzazione del 3D per meglio rappresentare strutture di accesso e la natura dell'oggetto o dell'informazione cercata.

Partendo dunque da un modello di struttura e di navigazione noto, specificato in HDM, e su cui già è presente un sito web di informazioni, una parte del nostro lavoro si è rivolta a fornire sistemi di generazione automatica di strutture di accesso tridimensionali, che consentissero di realizzare proprio questo aspetto: un nuovo strato di collaborazione potenziata che sostituisse ricerche potenzialmente infruttuose mediante parole chiave.

L'applicazione sviluppata all'interno della ricerca, il Configuratore 3D, consente di caricare una progettazione HDM, realizzata mediante l'editor HDM e formalizzata in un file XML pronto per essere usato all'interno del framework Jweb per la generazione del sito bidimensionale, e di usarla, attraverso alcuni passaggi di mappatura, per la generazione del corrispondente ambiente tridimensionale di accesso

---

<sup>16</sup> Si tratta del maggiore sistema commerciale di collaborazione assistiva bidimensionale, con ottime funzionalità: <http://www.liveperson.com>

#### 4.4 Configuratore 3D

Il configuratore 3D si colloca nella catena operativa di sviluppo dell'applicazione tridimensionale tra la progettazione HDM, da effettuare mediante l'HDM Editor, e il *deploy* del mondo tridimensionale. Esso consente dunque non solo di creare una struttura di accesso 3D a partire da un progetto già esistente, ma eventualmente di specificare da zero una nuova struttura di accesso (senza partire da una progettazione HDM a priori), o eventualmente integrare una struttura generata per traduzione del progetto HDM. Un altro aspetto aggiunto al configuratore, per completare la possibilità di specificare le caratteristiche di fruizione dell'ambiente tridimensionale, è stato quello di offrire un'interfaccia per specificare le metafore cooperative (nella loro versione di base) di accesso al mondo virtuale.

In generale le funzioni di configurazione del mondo virtuale di accesso si suddividono in tre aree:

- *Istanziamento dei tipi*: è una fase che consente di far corrispondere, attraverso una mappatura, uno schema progettato in HDM con categorie di oggetti tridimensionali preposti a rappresentare il medesimo concetto di accesso. Ad esempio, componenti HDM che modellano il link a diverse entità, potrebbero essere mappate con geometrie rappresentanti porte tridimensionali, e recanti un'etichetta descrittiva.
- *Generazione diretta del mondo*: consente di specificare direttamente la struttura del mondo virtuale, a prescindere da una progettazione HDM a priori, ed esprimendo la struttura di accesso direttamente in termini degli oggetti e dei componenti del mondo virtuale
- *Specifiche delle metafore cooperative*: specificare quali elementi cooperativi (nella versione base del modello di cooperazione) attivare per l'accesso al mondo.

Risulta chiaro come l'aspetto concettualmente più complesso stia nella mappatura della terminologia generica di accesso HDM sulla metodologia strutturale adottata dai mondi virtuali.

I componenti del mondo virtuali sono raggruppati essenzialmente in modo gerarchico, utilizzando i concetti di *Mondo*, *Parte*, *Template*, *Oggetto*, e caratteristiche (o attributi) di ciascun oggetto.

Si definisce *Mondo* l'insieme completo di tutte le entità che formano l'ambiente virtuale di accesso all'applicazione e ai contenuti. In esso gli avatar si incontrano, navigano ed interagiscono secondo le modalità dettate sia dalla topologia del mondo (e quindi dai suoi aspetti geometrici), sia dalle sue caratteristiche dinamiche (animazioni e comportamenti degli oggetti, programmazione delle metafore di cooperazione). Poiché molto spesso un mondo risulta molto articolato, ospitando un gran numero di oggetti o di ambienti con finalità diverse, generalmente (sia per motivi concettuali che implementativi) si scompone un Mondo completo in sottoambienti definiti *Parti*. I visitatori di un mondo possono passare da una parte all'altra, evento che può in alcuni casi comportare differenze nelle modalità di collaborazione tra avatar, oppure nel modo

di rappresentare gli oggetti. Le Parti sono dunque l'unità minima spaziale che definisce l'ambito dello spazio condiviso in cui gli avatar possono cooperare. Le Parti sono costituite da una ben precisa geometria, e fungono da contenitore per un certo numero di *Oggetti*, dotati anch'essi di una geometria tridimensionale, disposti all'interno della parte a definire ulteriormente le caratteristiche dello spazio condiviso. Più oggetti possono condividere alcune caratteristiche – ad esempio avere la medesima geometria, con solo alcuni attributi variati. E' dunque possibile pensare che più oggetti siano formati derivando da un medesimo *Template*. Ad esempio, per rappresentare in un ambiente un certo numero di bottiglie con diverse etichette, ciascun oggetto bottiglia può derivare dal generico template della bottiglia, ma applicare sulla comune geometria una *texture map* differente a rappresentare una diversa etichetta. Gli *Attributi* di un oggetto rappresentano le diverse caratteristiche che sono attribuibili ad un oggetto stesso. Gli attributi principali sono:

- *Aspetto*: riguarda il materiale (o la texture mapping) applicabile alla rappresentazione dell'oggetto
- *Geometria*: riguarda la collezione di vertici e poligoni che compongono la rappresentazione puramente tridimensionale dell'oggetto
- *Comportamento (Behavior)*: rappresenta la modalità con cui questo oggetto interagisce con il mondo in cui è innestato oppure con gli utenti, ad esempio se è dotato di un'animazione, se reagisce al click dell'utente ed in che modo, etc.
- *Posizione*: posizione dell'oggetto nella parte.
- *Condivisione*: specifica se un oggetto può essere manipolato o meno dagli avatar che sono ubicati nella sua stessa parte
- *Teletrasporto*: permette fondamentalmente di specificare una eventuale parte nella quale deve essere proiettato un avatar quando collide con un oggetto.

Particolarmente interessante è la possibilità di lavorare con le *appearance* (aspetto esteriore). Infatti queste consentono di riutilizzare più volte una stessa geometria caratterizzandola tramite una diversa visualizzazione.

Supponendo di lavorare con una geometria che rappresenti una bottiglia, possiamo considerare l'*appearance* come la texture da usare per ricoprire l'oggetto. Risulta evidente quindi che sostituendo la sola *appearance* e senza modificare la geometria è possibile dare la sensazione all'utente di interagire con oggetti diversi. E' possibile infatti usare sia una texture trasparente per dare l'idea di una bottiglia vuota che una texture scura per dare l'idea di una bottiglia piena.

*Animazioni e Behavior* permettono di rendere interattivo il mondo: le animazioni risultano particolarmente utili per programmare azioni non eccessivamente complesse e prive di interazioni con l'utente; i *behaviour* sono invece molto più potenti delle animazioni. Questi infatti consentono di stabilire la reazione di un oggetto al particolare comportamento dell'utente. Sia *animazioni* che *behaviour* sono programmabili a priori utilizzando opportuni linguaggi di programmazione (v. oltre sull'uso di Java3D).

Infine il *teletrasporto* consente agli utenti di muoversi tra le parti. Infatti è possibile associare ad un oggetto la proprietà teletrasporto e la parte in cui l'utente deve essere proiettato. Quando l'avatar collide con questo oggetto viene spostato nella parte specificata. E' inoltre possibile specificare, invece della parte, anche una destinazione

*http*. Operando in questo modo la reazione alla collisione sarà l'apertura di un browser HTML contenente la pagina associata all'oggetto tridimensionale.

Queste definizioni riguardano aspetti strutturali dell'ambiente virtuale di accesso. Quale sia l'effettivo valore semantico in relazione al tipo di navigazione e al tipo di struttura di accesso che questi componenti possono realizzare, dipende da una mappatura concettuale che deve essere di volta in volta fatta, tra la struttura di accesso HDM e la struttura di accesso 3D realizzabile attraverso questi componenti.

In altre parole, non esiste a priori una corrispondenza fissa tra un concetto HDM ed un concetto 3D, ma è chiaramente possibile, seguendo la natura dell'applicazione che si vuole realizzare, decidere come effettuare questa mappatura.

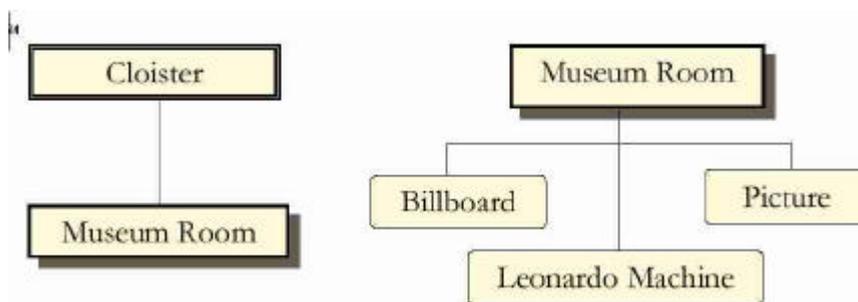
Per fissare le idee con un esempio, si può decidere di far corrispondere a un centro di collezione un determinato ambiente 3D (una stanza), le cui porte (oggetti 3D ottenuti dal template geometrico "Porta") corrispondono a link verso altre entità, le quali possono essere rappresentate da altri oggetti.

Il processo di derivazione della struttura di accesso tridimensionale a partire dalla progettazione HDM deve quindi seguire alcune fasi:

1. *Derivazione dell'Albero dei Tipi*. Essa è la struttura ad albero che visualizza la gerarchia della struttura progettata attraverso HDM. All'interno dell'albero dei tipi è possibile visualizzare i tipi di ciascuna voce (Entity, Component o Collection), con il relativo nome.
2. *Associazione ai componenti 3D*. A partire dall'albero dei tipi, occorre trovare una associazione concettuale tra le entità, i componenti, e le collezioni, e la rappresentazione che esse possono avere nella struttura 3D. Per effettuare tutte le successive operazioni su uno schema HDM è possibile selezionare una serie di filtri che consentono di visualizzare solo parte della struttura. È possibile, infatti, che lo schema sviluppato in HDM risulti piuttosto complesso e molto strutturato. Proprio a causa di questa limitazione è probabile che non sia particolarmente intuitiva l'opera di associazione parti/oggetti. Al fine di poter operare meglio quindi è possibile, tramite l'ausilio del filtro che permette di visualizzare le sole Entità, concentrare la propria attenzione su questo particolare costruito dello schema HDM. Quindi da uno schema logico che rappresenta la struttura di un sito, effettuando le associazioni viste precedentemente, è possibile ricavare una corrispondente vista iniziale per la generazione di un mondo virtuale.
3. *Valorizzazione*. Una volta definite le geometrie ed i template 3D da associare a ciascun componente o entità, occorre effettivamente indicare quante istanze per ciascun componente esistono effettivamente nell'hyperbase. A questo scopo si collega lo schema concettuale con un database contenente tutte le informazioni relative alle geometrie delle istanze e dei template. Questo procedimento è detto appunto valorizzazione, dato che assegna un effettivo valore di istanza al generico componente della struttura HDM. La valorizzazione comprende anche l'effettivo posizionamento degli Oggetti all'interno delle Parti e del Mondo.

4. *Definizione delle metafore cooperative.* Utilizzando un'apposita finestra del configuratore, si possono selezionare gli elementi cooperativi (appartenenti alla versione base del modello cooperativo) da accoppiare al mondo. I risultati di questa configurazione vengono descritti utilizzando un formato XML, interpretabile dal software di rappresentazione grafica della scena tridimensionale, e da quello che si occupa di gestire l'interazione tra gli utenti e la condivisione dello stato.
5. *Generazione automatica della struttura 3D.* Una volta definita la mappatura HDM-3D e definite le effettive valorizzazioni, il configuratore è in grado di generare un file XML che descrive come deve essere strutturato il mondo 3D. Questo file XML può essere prelevato da un visualizzatore, lo Scene Builder, che si occupa di disegnare in tempo reale la scena utilizzandone le informazioni.

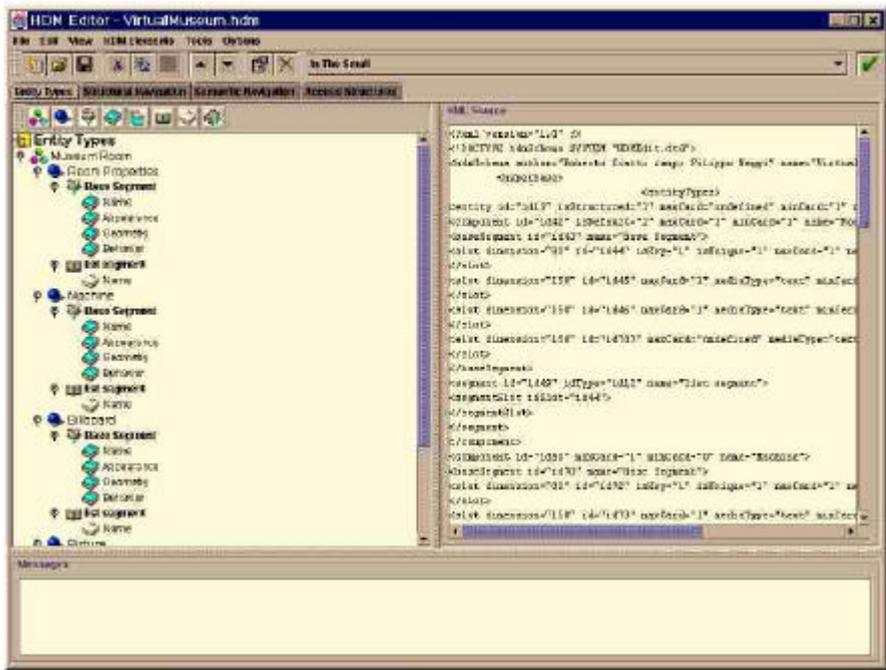
Per portare un esempio, si supponga di voler modellare l'applicazione "Leonardo Virtuale"<sup>17</sup>, che è stata utilizzata come caso di studio per numerosi sottoprototipi della ricerca. In questa applicazione tridimensionale, si disponeva di un sito convenzionale, composto da una pagina centrale, da cui era possibile accedere a sottopagine web, ciascuna corrispondente alla descrizione di una delle macchine concepite da Leonardo da Vinci. Ciascuna pagina bidimensionale comprendeva una foto del disegno leonardesco, ed una descrizione testuale del comportamento della macchina. In HDM, dunque, è possibile identificare una *collezione*, denominato "chostro del museo", che utilizza come *centro* la pagina principale di accesso alle sottopagine. Le sottopagine rappresentano delle *entità*, la "stanza del museo", composte da tre componenti, un'immagine e una descrizione testuale. Nella rappresentazione tridimensionale, la pagina web, che idealmente rappresenta la "stanza del museo", è composta invece da tre componenti: un'immagine bidimensionale, un modello tridimensionale della macchina di Leonardo, ed un cartello che fornisce il collegamento alla corrispondente pagina bidimensionale.



**Fig. 4-1: Lo schema HDM di Leonardo Virtuale. Il centro di collezione, il "chostro", porta a diverse "stanze di museo", ciascuna contenente una macchina di Leonardo, ed altri componenti**

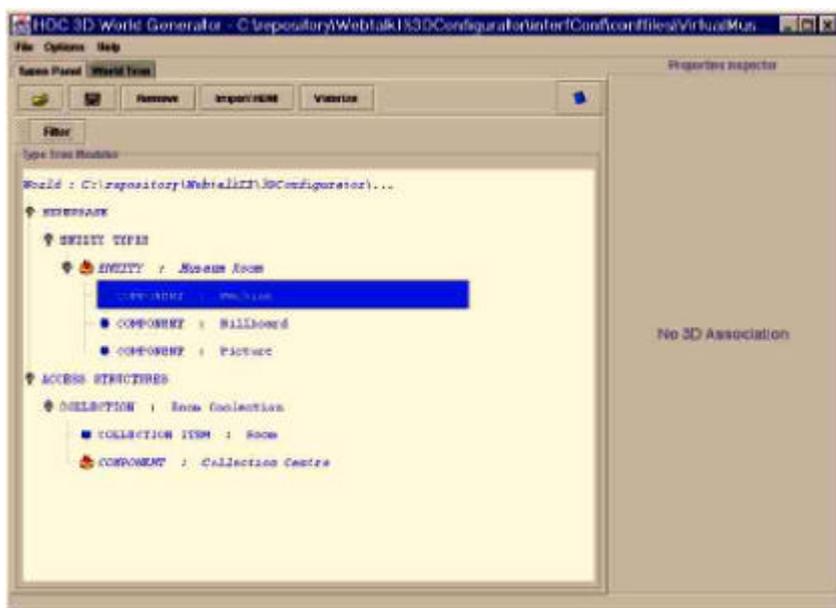
<sup>17</sup> op.cit.

Questa progettazione produce un albero di tipi, importabile dal configuratore, come segue:



**Fig. 4-2:** la progettazione HDM eseguita mediante l'editor HDM. Si può notare come ciascun componente, nell'ottica di essere trasformato in un corrispettivo 3D, sia dotato di Segmenti di tipo Nome, Aspetto, Geometria, Comportamento.

Ciascun componente progettato riceve dei Segmenti che alludono alla futura corrispondenza con la struttura tridimensionale: in particolare, ciascun segmento corrisponde ad un attributo rilevante assegnabile ad un oggetto: il suo nome, il suo aspetto, la sua geometria, ed il suo comportamento. La descrizione dei tipi viene generata in formato XML dall'HDM Editor, e risulta importabile nel configuratore 3D.



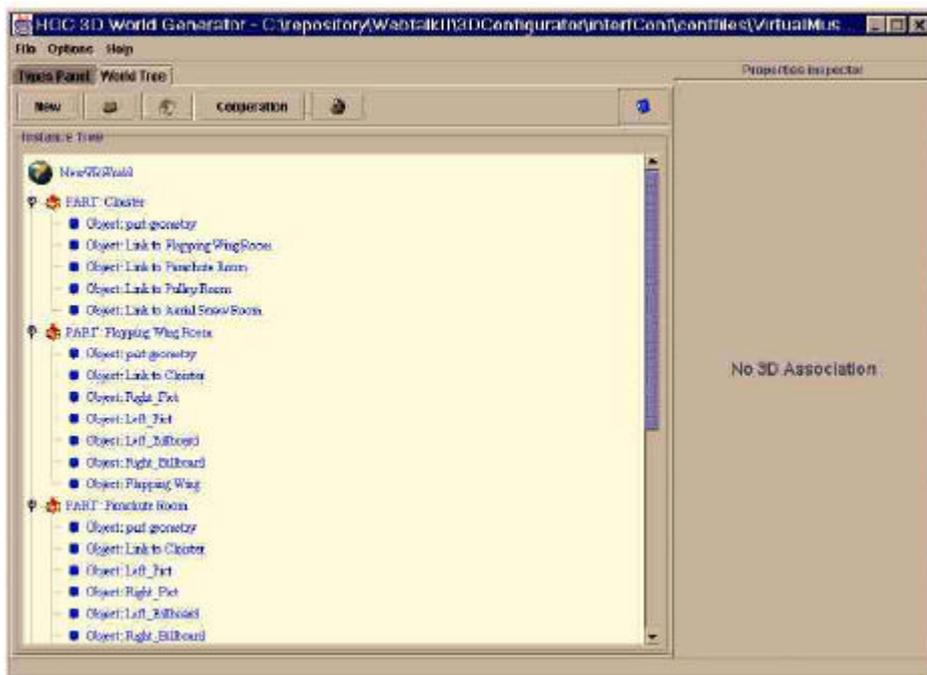
**Fig. 4-3: l'importazione dello schema HDM nell'albero dei tipi del configuratore.**

Nell'albero dei tipi del configuratore è possibile notare la separazione tra le entità che formano l'hyperbase (nell'esempio discusso, la base dei contenuti è formata dall'insieme delle entità Stanza di Museo, ciascuna delle quali contiene i componenti Macchina, Cartello e Immagine), e tra le strutture d'accesso. La struttura d'accesso descritta prevede l'uso di una collezione di stanze. L'oggetto collezionato è dunque la stanza, ed esiste un componente (che sarà un ambiente che rappresenta un cortile di raccordo tra tutte le stanze) che funge da centro di collezione.

A questo punto è possibile far corrispondere ai tipi HDM le rispettive entità strutturali 3D, mediante l'assegnazione dei template che rappresentano determinati componenti. Ad esempio al Componente Museum Room è possibile accoppiare il Template Stanza, che contiene la geometria della stanza, e le posizioni predefinite della Macchina, del Cartello e dell'Immagine. Quando nell'albero dei tipi è stato associato un template tridimensionale, la voce corrispondente dell'albero dei tipi viene marcata da un simbolo di "pacchetto", che indica la presenza di un template 3D in associazione.

Terminata l'associazione con i template tridimensionali, è possibile passare alla valorizzazione vera e propria, vale a dire imporre l'associazione tra ciascun componente e le  $n$  istanze di questo componente, presenti in un apposito database di oggetti, realizzato in un formato interpretabile dal software di rendering della scena, lo Scene Builder.

La fase di valorizzazione porta, nell'esempio, al seguente risultato:



**Fig. 4-4.** La fase di valorizzazione connette la struttura concettuale 3D con le effettive istanze presenti nel database delle geometrie.

In particolare, nell'esempio si è identificato il centro di collezione e ciascuna delle stanze come una Parte, in modo da ottimizzare i tempi di caricamento e la gestione della collaborazione tra gli utenti all'interno del mondo. Le stanze, nella valorizzazione, risultano essere quattro (la stanza "del paracadute", la stanza delle "ali battenti", la stanza del "battipalo", e la stanza della "vite aerea"), oltre al centro di collezione, rappresentato dalla geometria del Chiostro ispirato al cortile interno del Museo stesso. Per ciascuna singola istanza, il configuratore consente di impostare le varie proprietà, come ad esempio l'effettiva posizione dell'oggetto all'interno del mondo, l'aggancio ad un particolare comportamento (rappresentato da una classe Java, come verrà descritto più avanti), ecc.

Successivamente, attraverso una finestra offerta dal configuratore, si selezionano gli elementi di cooperazione di base che si desiderano applicare al mondo:



**Fig. 4-5.:** la finestra per la selezione degli elementi cooperativi, e la creazione di metafore cooperative. E' possibile collegare una particolare configurazione a ciascun gruppo di visita all'interno dello spazio collaborativo.

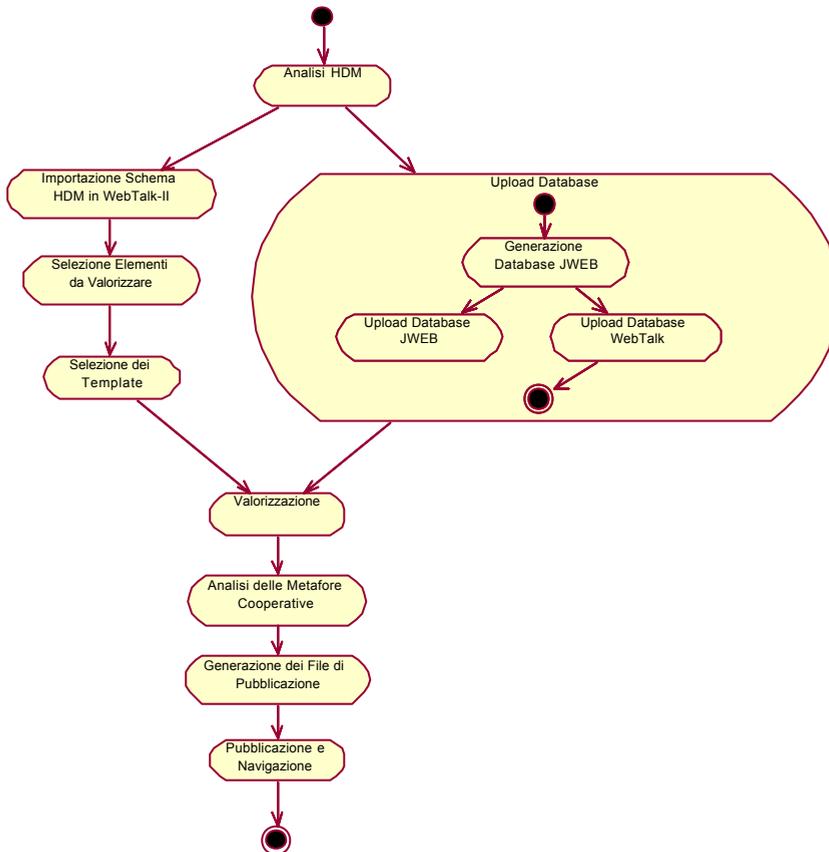
All'interno della finestra delle impostazioni collaborative, si possono creare più gruppi di utenti, e collegare a ciascun gruppo di utenti determinate impostazioni di collaborazione.

La fase finale prevede l'esportazione, in formato XML, delle informazioni strutturali del mondo 3D, ottenute dalla fase di valorizzazione, e la descrizione, sempre in formato XML, dei gruppi di collaborazione e dei relativi elementi cooperativi ad essi agganciati.

Lo schema seguente offre una panoramica del flusso di lavoro necessario ad ottenere una gestione e sincronizzazione dei contenuti tra un sito 2D ed una struttura di accesso 3D che offre su di esso uno strato di accesso collaborativo:

Per riassumere, la fase di progettazione e realizzazione dell'ambiente virtuale prevede in generale le seguenti attività, ciascuna di competenza di un ben preciso gruppo di persone:

- *Redazione dei contenuti e definizione degli obiettivi:* Si tratta della fase iniziale, in cui si definisce l'ambito dell'applicazione che si vuole sviluppare, che cosa essa vuole comunicare, quali sono i contenuti che devono essere oggetto dell'esplorazione e della collaborazione, e mediante quali meccanismi di accesso e di collaborazione devono essere fruiti. Queste decisioni sono normalmente a carico dell'esperto dei contenuti che commissiona la realizzazione dell'applicazione.
- *Design logico:* progettazione di entità, componenti, link strutturali e strutture di accesso secondo una politica HDM. Questa fase, come visto in precedenza, è comune anche al design del modello 2D della realtà da rappresentare. Preferibilmente va effettuata da un esperto di HDM che effettui un'accurata analisi della realtà da rappresentare.



**Fig. 4-6.:** il flusso di lavoro necessario alla generazione automatica di una struttura di accesso 3D su contenuti organizzati per un sito web 2D

- *Design grafico:* realizzazione di tutte le geometrie da utilizzare tramite uno strumento CAD per la creazione di oggetti virtuali. Questa parte dovrebbe essere svolta da un *modeler*, ovvero un esperto nell'utilizzo di strumenti per la grafica tridimensionale.
- *Specifiche proprietà:* definizione per ogni geometria delle proprietà che ne consentono la combinazione con altre geometrie. Deve essere svolta da una persona che conosca l'architettura di WebTalk-II e le funzionalità offerte dal configuratore 3D.

#### 4.5 Formati di pubblicazione e configurazione

L'utilizzo del formato XML risulta uno dei *leit-motiv* di tutta l'architettura generale del sistema collaborativo da noi pensato, a dimostrazione di come effettivamente esso possa essere usato come linguaggio generale per la rappresentazione di qualsiasi tipo di informazione strutturata, che sia necessario intercambiare tra più moduli funzionali. Nel caso particolare della configurazione dei mondi tridimensionali e delle modalità di collaborazione ad esso, XML viene pesantemente usato mediante la definizione di tre particolari DTD:

- Il DTD denominato *CML* (Configurator Markup Language), che consente di descrivere la struttura geometrica e dinamica del mondo, in termini di come gli oggetti sono disposti all'interno di una parte
- Il DTD denominato *MML* (Metaphor Markup Language), che descrive quali gruppi di utenti sono definiti per l'accesso ad un mondo, e quali elementi cooperativi di base possono da essere utilizzati
- Il DTD denominato *WSL* (World Specification Language), che descrive come sono collegate tra di loro le parti e quali metafore ciascuna parte usa. In altre parole esso funge da descrittore globale dell'applicazione, definendo la rete di CML (ciascun CML corrispondendo a una parte) e la relazione tra un CML ed il rispettivo MML.

La definizione del DTD relativo al CML è la seguente:

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Part (Initial_pos?, Object+)>
<!ELEMENT Initial_pos EMPTY>
<!ELEMENT Object (Geometry, Pos, Behavior?, Object*)>
<!ELEMENT Geometry (Location, Appearance?)>
<!-- Tag Inside Geometry -->
<!ELEMENT Location EMPTY>
<!ELEMENT Appearance EMPTY>
<!-- ***** -->
<!ELEMENT Pos EMPTY>
<!ELEMENT Behavior (Inter*, Anim*, Tele?)>
<!-- Tag Inside Behavior -->
<!ELEMENT Inter EMPTY>
<!ELEMENT Anim EMPTY>
<!ELEMENT Tele EMPTY>
<!-- ***** -->
<!-- Attribute of the elements listed above -->
<!ATTLIST Part Name CDATA #REQUIRED>
<!ATTLIST Initial_pos X CDATA #REQUIRED
Y CDATA #REQUIRED
Z CDATA #REQUIRED
angleX CDATA #REQUIRED
angleY CDATA #REQUIRED
angleZ CDATA #REQUIRED
>
<!ATTLIST Object Name CDATA #REQUIRED
id ID #REQUIRED
shared (Yes|No) "No">
<!ATTLIST Location URL CDATA #REQUIRED>
```

```

<!ATTLIST Appearance URL CDATA #REQUIRED>
<!ATTLIST Pos X CDATA #REQUIRED
Y CDATA #REQUIRED
Z CDATA #REQUIRED
angleX CDATA #REQUIRED
angleY CDATA #REQUIRED
angleZ CDATA #REQUIRED
>
<!ATTLIST Inter Class CDATA #REQUIRED>
<!ATTLIST Anim Class CDATA #REQUIRED>
<!ATTLIST Tele Type (CML|HTTP|FTP) "CML"
URL CDATA #REQUIRED
ObjRef IDREF #IMPLIED>

```

**Fig. 4-7: Il DTD relativo al CML. Il Configuration Markup Language consente di descrivere come gli oggetti 3D sono montati in una Parte dell'ambiente tridimensionale.**

Come si può osservare, l'*elemento radice* è rappresentato dal tag *part*, che ammette un attributo *Name* e dei sotto-elementi *Object* ed *initial\_pos*. Quest'ultimo elemento è opzionale, mentre vi possono essere uno o più oggetti costituenti una parte; l'oggetto di *default* è, infatti, quello che rappresenta la geometria stessa della parte. Gli oggetti, che possono, a loro volta, contenere altri oggetti, sono caratterizzati da varie proprietà, alcune delle quali sono espresse sfruttando degli appositi "marcatori" che prendono il nome di *Empty Tag*. Questi tag hanno valore per gli attributi che portano con sé, in quanto non possono avere un contenuto. Gli attributi di un elemento sono descritti in un'apposita lista in cui viene precisato anche il tipo di ciascuno di essi.

Di seguito viene riportato, a titolo di esempio, un CML generato automaticamente da una delle parti ottenute per valorizzazione durante l'esempio del flusso di lavoro di gestione dei contenuti dell'applicazione "Leonardo Virtuale":

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Part SYSTEM "..\CML.dtd">
<Part Name="Aerial Screw Room">
<Initial_pos X="0.0" Y="0.0" Z="0.0" angleX="0.0" angleY="0.0" angleZ="0.0"/>
<Object Name="Room Geom" id="idVrMuseum1" shared="No">
<Geometry>
<Location URL="K://WebTalk2//Geo//Sala-A.wr1"/>
</Geometry>
<Pos X="0.0" Y="0.0" Z="0.0" angleX="0.0" angleY="0.0" angleZ="0.0"/>
</Object>
<Object Name="Aerial Screw" id="idVrMuseum2" shared="No">
<Geometry>
<Location URL="null"/>
</Geometry>
<Pos X="0.0" Y="0.0" Z="4.0" angleX="0.0" angleY="0.0" angleZ="0.0"/>
</Object>
<Object Name="left_bill" id="idVrMuseum3" shared="No">
<Geometry>
<Location URL="null"/>

```

```

</Geometry>
<Pos X="3.0" Y="1.0" Z="0.0" angleX="0.0" angleY="0.0" angleZ="0.0"/>
<Behavior>
<Tele Type="HTTP" URL="http://webtalk.elet.polimi.it/main.htm"
ObjRef=""/>
</Behavior>
</Object>
<Object Name="right_bill" id="idVrMuseum4" shared="No">
<Geometry>
<Location URL="null"/>
</Geometry>
<Pos X="-3.0" Y="1.0" Z="0.0" angleX="0.0" angleY="0.0" angleZ="0.0"/>
</Object>
<Object Name="left_picture" id="idVrMuseum5" shared="No">
<Geometry>
<Location URL="null"/>
</Geometry>
<Pos X="5.0" Y="7.0" Z="0.0" angleX="0.0" angleY="0.0" angleZ="0.0"/>
</Object>
<Object Name="right_picture" id="idVrMuseum6" shared="No">
<Geometry>
<Location URL="null"/>
</Geometry>
<Pos X="-5.0" Y="7.0" Z="0.0" angleX="0.0" angleY="0.0" angleZ="0.0"/>
</Object>
<Object Name="door" id="idVrMuseum7" shared="No">
<Geometry>
<Location URL="null"/>
</Geometry>
<Pos X="0.0" Y="2.0" Z="-5.0" angleX="0.0" angleY="0.0" angleZ="0.0"/>
<Behavior>
<Tele Type="CML" URL="http://hoc115.elet.polimi.it/maincml.cml"
ObjRef=""/>
</Behavior>
</Object>
</Part>

```

**Fig. 4-8 Esempio di file CML generato per l'applicazione "Leonardo Virtuale"**

Tutti i file relativi alle varie "parti", che costituiscono il mondo virtuale, vengono archiviati in una directory appartenente ad un repository, a cui i client possono accedere per costruire il grafo della scena.

Nel formato MML (Metaphor Markup Language), invece, vengono precisati i gruppi predefiniti e le loro caratteristiche, nonché l'abilitazione o meno, alla creazione di gruppi dinamici. Il server sfrutta queste informazioni allo scopo di gestire coerentemente i client, salvaguardando le impostazioni volute dal designer del mondo tridimensionale.

Di seguito, riportiamo il *DTD* contenente le specifiche del file *MML*:

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Cooperation (Group*)>
<!ELEMENT Group
(Metaphor,Motion,Relation,Chat,Audio,Visualization,SharedState)>
<!ELEMENT Metaphor (Protection*,Distruccion*)>
<!ELEMENT Motion (RealityLike*,ZoneJump*,NearJump*)>
<!ELEMENT Relation (Tether*,Link*,Use*)>
<!ELEMENT Tether (Single*,Multiple*,Falling*,FanIn*,FanOut*)>
<!ELEMENT Link (Single*,Multiple*,Falling*,FanIn*,FanOut*)>
<!ELEMENT Use (UseSingle*,UseMultiple*)>
<!ELEMENT Chat (Private*,Free*,Groupped*,RestrictedGroup*,Whisper*,Ext*)>
<!ELEMENT Audio (HalfDuplex*,FullDuplex*)>
<!ELEMENT Visualization (Scene*,Pow*)>
<!ELEMENT Scene
(SceneGlobal*,SceneGroupped*,SceneSelection*,SceneSelf*)>
<!ELEMENT Pow (Normal*,Next*,Panoramic*,Outer*,OtherAvatar*)>
<!ELEMENT SharedState (ShareGlobal*,ShareGroupped*,Time*,Space*)>
<!ELEMENT Space EMPTY>
<!ELEMENT Time EMPTY>
<!ELEMENT Protection (Kicking*,Banning*,FixedSize*,Free*,Password*)>
<!ELEMENT Kicking EMPTY>
<!ELEMENT Banning EMPTY>
<!ELEMENT FixedSize EMPTY>
<!ELEMENT Free EMPTY>
<!ELEMENT Password EMPTY>
<!ELEMENT Distruccion (Persistent*,Dynamic*,Forced*)>
<!ELEMENT Persistent EMPTY>
<!ELEMENT Dynamic EMPTY>
<!ELEMENT Forced EMPTY>
<!ELEMENT RealityLike EMPTY>
<!ELEMENT ZoneJump EMPTY>
<!ELEMENT NearJump EMPTY>
<!ELEMENT Single EMPTY>
<!ELEMENT Multiple EMPTY>
<!ELEMENT Falling EMPTY>
<!ELEMENT FanIn EMPTY>
<!ELEMENT FanOut EMPTY>
<!ELEMENT UseSingle EMPTY>
<!ELEMENT UseMultiple EMPTY>
<!ELEMENT Private EMPTY>
<!ELEMENT Groupped EMPTY>
<!ELEMENT RestrictedGroup EMPTY>
<!ELEMENT Whisper EMPTY>
<!ELEMENT Ext EMPTY>
<!ELEMENT HalfDuplex EMPTY>
<!ELEMENT FullDuplex EMPTY>
<!ELEMENT SceneGlobal EMPTY>
<!ELEMENT SceneGroupped EMPTY>
<!ELEMENT SceneSelection EMPTY>
<!ELEMENT SceneSelf EMPTY>
<!ELEMENT Normal EMPTY>
<!ELEMENT Next EMPTY>
<!ELEMENT Panoramic EMPTY>
<!ELEMENT Outer EMPTY>
<!ELEMENT OtherAvatar EMPTY>
<!ELEMENT ShareGlobal EMPTY>
```

```

<!ELEMENT ShareGrouped EMPTY>
<!ATTLIST Group Name CDATA #REQUIRED>
<!ATTLIST FixedSize SIZE CDATA #REQUIRED>
<!ATTLIST Password WORD CDATA #REQUIRED>

```

**Fig. 4-9: Documento di specifica dei file relativi alle metafore cooperative**

La maggior parte dei *flag* che determinano le metafore cooperative, viene riportata nel file sfruttando il formalismo, precedentemente descritto, degli *empty* tag. La presenza del tag relativo ad una particolare proprietà, quindi, fornisce le indicazioni necessarie sia al server per l'abilitazione delle politiche di gestione della cooperazione, sia al client per l'impostazione della corretta interfaccia grafica.

Riportiamo di seguito un esempio ottenuto specificando alcuni gruppi di visita al museo Leonardo Virtuale, ed i relativi elementi collaborativi, nella versione base del modello di cooperazione.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Cooperation SYSTEM "MML.dtd">
<Cooperation>
  <Group Name="Ricercatori">
    <Metaphor>
      <Protection>
        <Kickin g/>
        <Bannin g/>
        <Free/>
      </Protection>
      <Distruction>
        <Persistent/>
      </Distruction>
    </Metaphor>
    <Motion>
      <RealityLike/>
    </Motion>
    <Relation>
      <Tether>
        <Single/>
        <FanIn/>
      </Tether>
      <Link>
        <Single/>
        <FanIn/>
      </Link>
    </Relation>
    <Use>
      <UseSingle/>
    </Use>
  </Group>
  .....
  <Visualization>
    <Scene>
      <SceneGlobal/>
      <SceneGrouped/>
    </Scene>
    <Selection/>
    <Self/>
  </Visualization>

```

```

</Scene>
<Pow>
<Normal/>
<Next/>
<Panoramic/>
<Outer/>
<OtherAvatar/>
</Pow>
</Visualization>
<SharedState>
<ShareGlobal/>
</SharedState>
</Group>
</Cooperation>

```

**Fig. 4-10: Esempio di file MML per l'accesso all'applicazione "Leonardo Virtuale"**

Il terzo DTD, come anticipato, è necessario per descrivere le relazioni strutturali tra i vari CML, ciascuno costituente una Parte, e descrivere dunque la topologia strutturale complessiva del mondo.

Questa informazione viene utilizzata a lato server per decidere, volta per volta, quale CML passare al renderer in tempo reale della scena collaborativa (Scene Builder).

La struttura è la seguente:

```

<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT world (cml_src, part+)>
<!ELEMENT cml_src (#PCDATA)>
<!ELEMENT part (obj*)>
<!ELEMENT obj (pos)>
<!ELEMENT pos EMPTY>
<!-- ***** -->
<!ATTLIST world name CDATA #REQUIRED>
<!ATTLIST part name CDATA #REQUIRED
isDefault (true|false) "false">
<!ATTLIST obj name CDATA #REQUIRED>
<!ATTLIST pos X CDATA #REQUIRED
Y CDATA #REQUIRED
Z CDATA #REQUIRED
angleX CDATA #REQUIRED
angleY CDATA #REQUIRED
angleZ CDATA #REQUIRED>

```

**Fig. 4-11 Proprietà del documento WSL, necessario alla definizione della rete topologica globale dei CML**

La radice del documento è costituita dall'elemento *World*, che contiene l'elenco delle "parti" (rappresentate dal tag *part*) e la direttiva *cml\_src*, necessaria per specificare l'indirizzo del repository dei file *CML*. Ciascun tag *part* possiede un attributo, *isDefault*, in grado di indicare al

server qual è la parte predefinita cui il client deve accedere nel caso non ne specifichi una in particolare: è questo il caso di un passaggio da un sito bidimensionale alla corrispondente shell tridimensionale.

Anche in questo caso, riportiamo un esempio relativo alla generazione automatica dell'applicazione "Leonardo Virtuale":

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE world SYSTEM "WorldStructure.dtd">
<world name="Museum">
  <cml_src>http://hoc.elet.polimi.it/cml</cml_src>
  <part name="Cloister" isDefault="true">
    <obj name="Machine">
      <pos X="5.0" Y="3.0" Z="0.0" angleX="0.0" angleY="0.0" angleZ="0.0"/>
    </obj>
  </part>
  <part name="Aerial Screw Room">
    <obj name="Aerial Screw">
      <pos X="6.0" Y="4.0" Z="0.0" angleX="0.0" angleY="0.0" angleZ="0.0"/>
    </obj>
    <obj name="Left Billboard">
      <pos X="3.0" Y="4.0" Z="0.0" angleX="0.0" angleY="0.0" angleZ="0.0"></pos>
    </obj>
    <obj name="Right Billboard">
      <pos X="9.0" Y="4.0" Z="0.0" angleX="0.0" angleY="0.0" angleZ="0.0"></pos>
    </obj>
  </part>
</world>
```

**Fig. 4-12: Esempio di file WSL relativo all'applicazione "Leonardo Virtuale"**

## 4.6 Conclusioni

Il linguaggio XML costituisce una eccellente opportunità per risolvere il problema della mappatura tra accesso e navigazione bidimensionale e lo strato di collaborazione tridimensionale.

Da concetti di progettazione dei contenuti, è stato possibile discendere alla definizione geometrica del mondo tridimensionale collaborativo con una struttura adatta alla fruizione dei contenuti stessi, mediante una definizione prima astratta, a livello di classi di oggetti, e successivamente attraverso una istanziazione, che prevede il collegamento di ciascuna classe ad una categoria di oggetti o a oggetti ben precisi.

Ne risulta la possibilità di progettare a priori una organizzazione di contenuti fruibili sia nello spazio bidimensionale, in modo solitario da parte degli utenti, sia in uno spazio tridimensionale, dove agli utenti sono offerte rappresentazioni più vivide, e la possibilità di interagire tra di loro.

## 5 La Rappresentazione Grafica Tridimensionale

### 5.1 *Premessa*

La realizzazione di un motore grafico che consenta la rappresentazione di un ambiente di collaborazione costituisce uno dei punti nodali dell'architettura del nostro sistema di collaborazione. Desiderando dimostrare che la rappresentazione tridimensionale consente di rafforzare la collaborazione corroborando la percezione dello stato condiviso, è importante trovare una soluzione implementativa che consenta un'integrazione il più semplice possibile all'interno del lavoro, sia dal punto di vista concettuale (che consenta cioè di progettare facilmente il renderer grafico in relazione ai requisiti imposti dal contesto applicativo) che dal punto di vista effettivamente implementativo (difficoltà di scrittura del codice).

Oltre alla possibilità, già esplorata attraverso l'architettura WebTalk-I, di utilizzare un plug-in a cui affidare la renderizzazione di geometrie modellate attraverso un formato standard (es. VRML), la scrittura diretta di un sottosistema di rendering consente di ottenere una maggiore libertà nell'implementare funzionalità altrimenti non possibili attraverso il plugin. Per citare un esempio, utilizzando il solo paradigma di modellazione VRML, le possibilità di interazione con la scena tridimensionale sono limitate all'uso dei sensori forniti da VRML. Si potrebbe tentare di estendere le modalità di interazione utente-scena, e in generale i comportamenti possibili degli oggetti della scena mediante la creazione di un'apposita struttura programmatica (il concetto di Behavior che verrà esposto a breve). Tra le numerosi API grafiche a disposizione per la scrittura di un renderer con queste caratteristiche, la scelta è ricaduta su Java3D.

Se dunque un filone della ricerca nell'ambito della rappresentazione grafica ha riguardato il modo migliore per utilizzare il paradigma ad oggetti per rappresentare eventi e necessità date da una scena che cambia dinamicamente in un contesto collaborativo, dall'altra è sorto il problema di eseguire queste operazioni con un livello di prestazioni accettabile. Su questo fronte, si sono condotti alcuni studi provando l'implementazione di alcuni algoritmi di ottimizzazione del rendering geometrico, utilizzando direttamente un renderizzatore scritto in OpenGL.

### 5.2 *Il Kernel Grafico Java 3D*

#### 5.2.1 **Caratteristiche di Java 3D**

La scelta di una API grafica quale Java 3D consente di ottenere in modo immediato vantaggi in termini di portabilità, modularità e flessibilità nella programmazione. Sfruttando le caratteristiche architettoniche di Java 3D, risulta possibile, in modo relativamente semplice, realizzare un'implementazione tecnica di alcuni dei concetti

fondanti della nostra struttura per sistemi di collaborazione virtuale, quali la renderizzazione dinamica della scena, il suo aggiornamento in tempo reale, e la sua costruzione mediante aspetti, comportamenti, e animazioni definiti in fase di progettazione: Java 3D infatti fornisce classi dalle quali è possibile istanziare oggetti creati apposta per la grafica 3D, che vanno dalle primitive necessarie al disegno (linee, punti, poligoni...) a classi per la definizione di comportamenti da associare agli oggetti.

L'approccio è quello di fornire una architettura ad oggetti che descriva le problematiche di trattamento della grafica 3D, realizzando la possibilità di analizzare e realizzare un problema di natura grafica utilizzando tutta la flessibilità e la potenza di un paradigma ad oggetti. La libreria si appoggia poi direttamente ad uno strato OpenGL oppure DirectX, consentendo di mantenere comunque un ragionevole livello prestazionale.

L'interposizione di uno strato ad oggetti tra l'effettivo codice applicativo e la logica di controllo grafico a basso livello OpenGL implica che alcune scelte spiccatamente tecniche dal punto di vista grafico, quali la scelta della modalità di rendering dei poligoni, delle politiche di poligon culling e di selezione delle superfici visibili, gli algoritmi di collisione e di rilevamento di selezione da parte dell'utente (picking) non sono direttamente decidibili dal programmatore, che delega questi aspetti più 'tecnici' all'implementazione scelta dagli autori della API Java 3D.

Se dunque da una parte Java 3D ha caratteristiche eccezionali che si adattano perfettamente all'organizzazione di una applicazione collaborativa in termini delle funzionalità rivolte alla creazione dinamica dell'applicazione e alla gestione dinamica delle modalità di interazione collaborativa, dall'altra toglie un certo grado di controllo nella ricerca della pura prestazione tecnica necessaria ad ottenere una visualizzazione grafica fluida, e gestita in modo razionale ed ottimizzato.

Due tuttavia sono i punti nodali per cui l'organizzazione ad oggetti di Java 3D offre una soluzione interessante per la realizzazione del kernel grafico collaborativo: la struttura di un Grafo di Scena, che descrive la situazione delle geometrie, delle caratteristiche e delle animazioni degli oggetti correntemente renderizzati a video per l'utente, e il concetto di Behavior e di Stimoli, che consentono la creazione di un sistema di gestione di eventi e di risposte totalmente programmabile ad oggetti.

Un grafo di scena Java 3D consiste essenzialmente in un *grafo aciclico diretto* (DAG) i cui nodi rappresentano gli oggetti che popolano il mondo e le loro caratteristiche come geometrie, suoni, luci, posizione, orientamento e aspetto visivo (texture, materiale, riflessione...).

Tutti i nodi sono sottoclassi di una classe astratta detta *SceneGraphObject*, che definisce metodi comuni. Una volta che uno *SceneGraphObject* è stato istanziato e connesso ad altri oggetti *SceneGraphObject* formando un sottografo, l'intero sottografo può essere attaccato alla radice dello *SceneGraph*, detta *VirtualUniverse*, attraverso un ulteriore oggetto detto *Locale*: questa fase rende 'vivo' il sottoramo del grafo di scena – in altre parole passa la gestione della renderizzazione al motore e agli algoritmi di rendering di

Java 3D, consentendo solo un'interazione limitata alle caratteristiche definite prima dell'entrata a runtime in scena dell'oggetto.

Gli oggetti derivati da *SceneGraphObject* possono essere infatti manipolati attraverso opportuni metodi di *set* e *get*, con una limitazione: le modifiche e le letture possono essere fatte solo durante la creazione del sottografo, con alcune eccezioni dove permesso. L'accesso alla maggior parte dei metodi di *set* e *get* di oggetti che fanno parte di uno sottografo "vivo" o compilato è vietata.

Tale restrizione permette al *Renderer*, l'oggetto di Java 3D che disegna la scena, di conoscere tutta l'informazione riguardante i nodi in anticipo, per ottimizzare la compilazione e la visualizzazione dello *SceneGraph*.

In alcuni casi è comunque permesso accedere ai *set* e *get* di oggetti che fanno parte di un sottografo già compilato, attraverso un insieme di *capability bits* che, abilitati, permettono l'utilizzo di particolari funzioni anche su nodi vivi.

Lo scene graph è costituito quindi da nodi in relazione padre-figlio, che formano una struttura ad albero partendo da una radice. La gerarchia degli oggetti che fanno parte del grafo è la seguente:

- *VirtualUniverse*
- *Locale*
- *SceneGraphObject*
- *Node*
- *Group*
- *Leaf*
- *NodeComponent*
- Diversi oggetti componenti
- *Transform3D*

Gli oggetti *Node* si dividono in *Group* e *Leaf*. I nodi *Group* possono puntare a 0 o n figli, ma possono avere un solo padre, e servono quindi a raggruppare i propri nodi figli in base alla semantica del gruppo.

I nodi *Leaf* non hanno figli ed hanno un solo padre. La loro funzione è quella di specificare gli elementi che Java 3D utilizza nelle fase di rendering, come geometria dell'oggetto 3D, luci, suoni...

Tra i gruppi distinguiamo *BranchGroups* (BG), ciascuno corrispondente ad un preciso sotto grafo e *TransformGroups* (TG), contenenti informazione sulle trasformazioni (rotazioni, traslazioni, scaling) applicabili agli oggetti 3D che stanno sotto di loro.

Un *BranchGroup* è, come detto, la radice di un sottografo della scena che può essere compilato come una unità ed attaccato al *VirtualUniverse*, oppure incluso come figlio in un nodo di tipo *Group* appartenente ad un altro sottografo.

Le seguenti operazioni possono essere eseguite su un *BranchGroup*:

Un *BranchGroup* può essere compilato. Se altri *BranchGroup* sono inclusi nel grafo, anche questi verranno compilati. Attraverso la compilazione, Java 3D crea una propria struttura interna ottimizzata del sotto-grafo che ha il *BranchGroup* come radice.

Un *BranchGroup* può essere inserito in un universo virtuale attaccandolo al *Locale*. L'intero sottografo viene quindi indicato come "vivo".

Un *BranchGroup* può essere staccato oppure associato ad altri sottografi a tempo d'esecuzione abilitando opportunamente alcuni permessi.

Quest'ultima proprietà è fondamentale: tutti gli oggetti 3D che popolano il mondo possono essere posizionati nell'universo virtuale a tempo di esecuzione, e quindi quando la fase di rendering è già iniziata per lo scene graph del mondo, solo se sono agganciati ad un *BranchGroup*.

Un *TransformGroup* specifica una singola trasformazione spaziale attraverso un oggetto *Transform3D*, permettendo così di posizionare, ruotare e scalare tutti i nodi figlio.

Il concatenamento delle trasformazioni di ogni *TransformGroup* in un cammino diretto da *Locale* fino ad una foglia definisce un modello di trasformazione composta che prendendo punti, normali e distanze dalle coordinate locali della foglia li trasforma in coordinate del mondo virtuale.

I nodi foglia definiscono entità atomiche come geometrie, luci e suoni, comportamenti, punto di vista, e possono contenere riferimenti a *NodeComponents*, come nel caso di oggetti *Shape3D*, caratterizzati, attraverso *NodeComponents*, da *geometry* e *appearance* precise.

Un particolare ramo del grafo di scena è costituito dal ramo di visualizzazione, che comprende un nodo denominato *Viewing Platform*. Alla *viewing platform* è possibile associare comportamenti e posizioni che influenzano il punto di vista, il campo di visualizzazione e le modalità di navigazione da parte dell'utente all'interno della rappresentazione data dall'altro lato del grafo di scena.

Un ulteriore vantaggio offerto dal concetto di Grafo di Scena, è che il formato originario che descrive la geometria da rappresentare non è vincolante. In ogni caso, infatti, deve esistere un processo di interpretazione (parsing) per interpretare la struttura del formato grafico di origine, ed utilizzarla per popolare e costruire il grafo di scena, utilizzando i componenti sopra descritti. Questo fa sì che esistano delle porzioni software intermedie, opportune classi dette *loaders*, attraverso le quali è possibile caricare una generica geometria definita in un qualsiasi formato comune di grafica 3D e tradurla nel corrispettivo Java 3D, con l'enorme vantaggio di separare fase di creazione dei contenuti (disegno di geometrie 3D) e fase di programmazione delle caratteristiche del mondo virtuale, come navigazione, luci, comportamenti associati agli oggetti, animazioni, suoni. Tra i formati più comuni che alcuni loader predefiniti consentono di caricare, vi sono 3DS (generato dal pacchetto di modellazione 3D Studio Max, VRML, X3D).

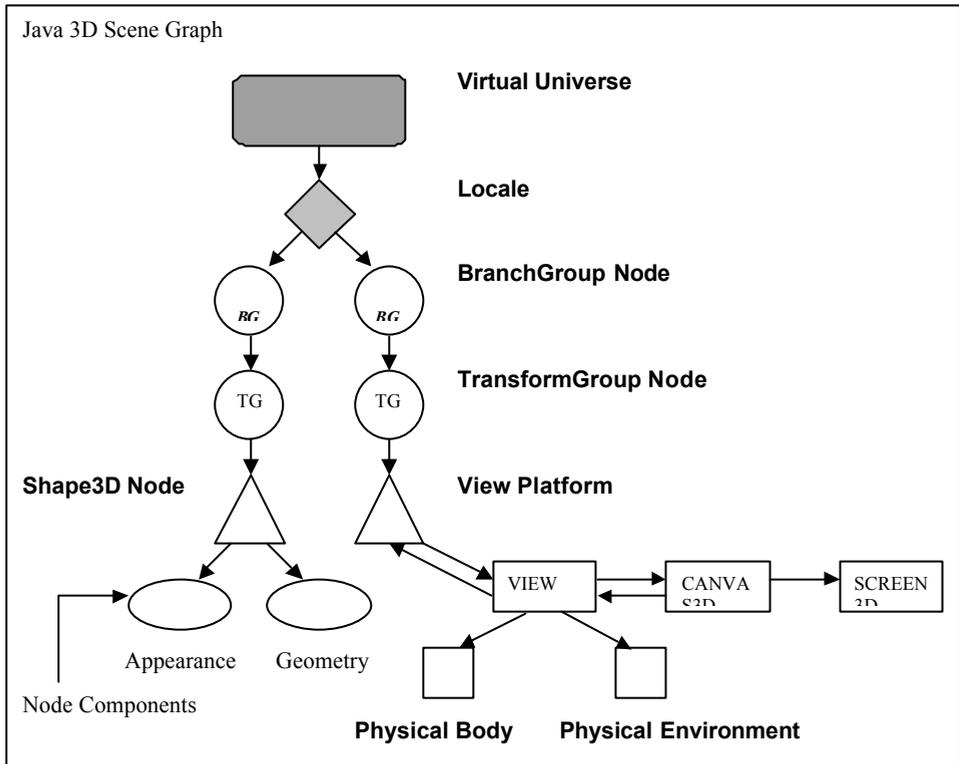


Fig. 5-1: la gerarchia degli oggetti Java 3D all'interno del Grafo di Scena

Il secondo aspetto di Java 3D riguarda, come detto, la possibilità di definire, all'interno del paradigma di progettazione ad oggetti, comportamenti e stimoli utilizzando un modello ad eventi.

A questo scopo esiste un nodo foglia detto *Behavior*, che fornisce i mezzi per animare gli oggetti, processare eventi di tastiera e mouse, reagire a movimenti effettuati nell'universo virtuale.

Behavior è infatti una classe astratta che contiene la struttura comune a tutti i comportamenti, come i due metodi *initialization* e *processStimulus*.

Il primo permette ad un comportamento di inizializzare il suo stato interno e specificare le condizioni di attivazione. Queste ultime sono implementazioni di una classe astratta *WakeupCondition*, che nella versione corrente di Java 3D è implementata in quattordici sottoclassi di *WakeupCriterion* e quattro classi che permettono la composizione mediante operatori logici (and, or, and di or e or di and) di più condizioni di sveglia.

Java 3D fornisce quindi un ricco insieme di criteri di attivazione (*WakeupCriterion*) che possono attivare un oggetto Behavior, ad esempio quando uno specifico

TransformGroup subisce cambiamenti, oppure si verifica uno specifico evento utente, come il click del mouse o la pressione di un tasto.

Il secondo metodo fondamentale della classe Behavior è il processStimulus, che costituisce l'azione eseguita dal comportamento e viene invocato quando il volume di attivazione della piattaforma visiva (lo spazio occupato dall'utente nello spazio) interseca il volume di attivazione del comportamento e le condizioni di sveglia sono verificate.

L'elaborazione degli eventi avviene solo all'interno di una zona spaziale di competenza, definita attraverso degli appositi limiti detti *bounds*. In questo modo si limita il carico computazionale necessario alla raccolta di tutti gli eventi e alla risposta secondo il comportamento programmato.

Inoltre, i behavior, i comportamenti, che possono essere associati agli oggetti 3D, possono essere scritti in Java, permettendo la creazione di un numero virtualmente infinito di possibili risposte di un oggetto ad un particolare stimolo.

### 5.2.2 Scene Builder: creazione dinamica degli universi virtuali

Tre elementi fondamentali della generica architettura per un sistema collaborativo da noi delinata sono stati, fino a questo momento, esposti:

- Una teoria generale della collaborazione, basata sul concetto della Metafora Cooperativa, e fornita in due versioni, una base, e l'altra avanzata
- La necessità di pensare gli ambienti virtuali come strutture di accesso collaborative, descrivibili mediante un generico linguaggio di progettazione (HDM), e descrivibili attraverso un linguaggio basato su XML
- La scelta di un linguaggio di programmazione grafico potente e basato sugli oggetti (Java 3D), che ha come cardini fondanti il concetto di Grafo di Scena e la gestione dei Behavior.

Queste premesse portano alla definizione di uno degli elementi chiave dell'architettura complessiva: il modulo di costruzione dinamica e presentazione della scena tridimensionale collaborativa. Questo modulo, denominato Scene Builder, deve poter essere in grado di leggere la descrizione CML della struttura tridimensionale, e di rappresentarla all'utente, fornendo a quest'ultimo la possibilità di collaborare mediante le modalità definite. Lo Scene Builder deve avere caratteristiche che consentano di realizzare i comportamenti di ciascun oggetto in modo modulare, e di interagire con un meccanismo che consenta di mantenere uno stato condiviso in tempo reale, in altre parole di dialogare con altre istanze dello Scene Builder, ciascuna rappresentante uno degli utenti partecipanti allo spazio condiviso. Quest'ultimo punto, che riguarda la gestione tecnica dello stato condiviso, implica la realizzazione di un modulo software a parte e la considerazione di numerosi aspetti tecnici.<sup>18</sup>

---

<sup>18</sup> Un interessante rassegna di tecniche e architetture implementative per la gestione tecnica dello stato condiviso si trova in Zyda M., Singhal S., *Networked Virtual Enviroments*, ACM Press, NewYork, 1999.

Partendo dal CML e dalla struttura tridimensionale che esso rappresenta, lo Scene Builder deve dunque gestire i seguenti aspetti del meccanismo di funzionamento del sistema collaborativo:

- *Collegamento dell'istanza locale della scena*, disegnata dallo Scene Builder locale, al sistema di mantenimento dello stato condiviso, tipicamente attraverso un middleware di rete, progettato per fornire le migliori prestazioni in termini di scalabilità (possibilità di supportare la collaborazione di un numero crescente di utenti) e consistenza (aggiornamento rapido dei cambiamenti imposti da ciascun utente nell'ambiente condiviso all'interno della propria istanza locale della rappresentazione dell'ambiente stesso)
- *Montaggio dinamico della scena tridimensionale* interpretando il linguaggio CML e caricando le geometrie richieste all'interno della finestra di visualizzazione dell'utente locale, dopo aver effettuato la fase di collegamento al sistema.
- *Ingresso e transito dell'utente locale e degli utenti remoti nelle "Parti"* (concetto definito a livello di CML), le sottounità strutturali di un "mondo", inteso come intera applicazione tridimensionale collaborativa. Il passaggio da una parte all'altra implica una serie di attività nell'aggiornamento dello stato condiviso in ciascuna delle rappresentazioni locali.
- *Aggiornamento dinamico di oggetti e avatar* durante la navigazione degli utenti, e aggiornamento del loro stato. Questa funzione è composta principalmente da due aspetti: una di *tracking* a livello locale, in cui lo Scene Builder è preposto a rilevare i movimenti e le interazioni con gli oggetti e i componenti del mondo eseguiti dall'utente locale nella propria istanza locale; ed un secondo aspetto di *updating*, in cui lo Scene Builder deve rappresentare nell'istanza locale movimenti ed aggiornamenti avvenuti in istanze remote della scena condivisa.
- *Gestione dei movimenti degli avatar*, che consiste nel notificare le istanze remote dei movimenti compiuti dall'utente nell'istanza locale, e nel rappresentare nell'istanza locale i movimenti degli altri utenti, ciascuno rappresentato da un'opportuna geometria, perlopiù antropomorfa, detta appunto "avatar".
- *Associazione di comportamenti agli oggetti*, vale a dire l'esecuzione in tempo reale di codice che descrive il cambiamento di stato di un oggetto (in termini di cambiamento delle sue proprietà o di innesco di altri eventi) in risposta ad un'interazione da parte dell'utente o di altri oggetti.
- *Filtraggio degli eventi* mediante interest management locale, che consiste nel selezionare istante per istante quali oggetti e quali eventi siano rilevanti per la rappresentazione corrente nell'istanza locale, omettendo di rappresentare ed elaborare tutte le informazioni non selezionate.
- *Gestione della navigazione e dei punti di vista*, che consiste nell'offrire funzionalità di navigazione nello spazio tridimensionale all'utente locale, e nel gestire la possibilità di visualizzare la scena utilizzando punti di vista diversi, non solo quello soggettivo (attraverso gli occhi dell'avatar dell'istanza locale), ma ad esempio attraverso "telecamere" predisposte all'interno dello spazio

condiviso, o attraverso il punto di vista di altri utenti presenti nello spazio condiviso.

Come lo Scene Builder gestisca questi aspetti, fondamentali per l'interazione basata su grafica 3D con l'utente, viene descritto nei paragrafi seguenti.

### **5.2.2.1 Collegamento al sistema**

Il collegamento (login) al sistema consiste nella richiesta da parte di un'opportuna interfaccia utente, di un nome, una password, e di un mondo o applicazioni entro cui l'utente desidera collaborare.

Una volta specificato il mondo di accesso, lo Scene Builder richiede il prelevamento della parte definita come punto di accesso iniziale. L'utente presceglie inoltre una geometria di avatar con cui desidera essere rappresentato.

La fase di connessione avviene con lo strato software preposto alla gestione dello stato condiviso, il quale controlla la disponibilità a gestire una collaborazione per la parte e per il mondo richiesto dall'utente. In caso positivo, lo strato di gestione dello stato condiviso aggiunge il nuovo utente alla propria lista di utenti.

Gli oggetti condivisi vengono dapprima creati e posizionati in determinati punti iniziali, secondo quanto definito nel CML. Tuttavia, l'ingresso dell'istanza locale potrebbe essere un caso di *latecomer*, vale a dire un utente che entra in uno spazio di lavoro condiviso in cui altri utenti hanno già modificato lo stato definito come iniziale. In questo caso, occorre aggiornare lo stato degli oggetti condivisi (il cui stato è modificabile mediante interazione da parte degli utenti) e rappresentare le posizioni degli avatar associati agli utenti già presenti. Queste operazioni vengono eseguite mediante l'applicazione di opportune trasformazioni dopo la creazione degli oggetti interessati nelle loro posizioni iniziali, durante la fase di montaggio della scena tridimensionale.

### **5.2.2.2 Montaggio della scena tridimensionale**

Il montaggio della scena tridimensionale avviene attraverso l'utilizzo di un pattern tripartito, con un componente Parser, un componente Mediator, ed il componente Scene Builder vero e proprio.

Il componente Parser si occupa direttamente della validazione del DTD cui appartiene il CML, e della sua interpretazione. Proseguendo progressivamente nella lettura dei nodi XML, al completamento di ciascun nodo rilevante e delle relative proprietà, viene sollevato un evento all'interno del componente Mediator. Il Mediator contiene una mappatura tra gli eventi sollevati dal Parser, e i metodi che devono essere invocati all'interno dello Scene Builder per la renderizzazione grafica effettiva a fronte dell'evento.

L'utilizzo di questo pattern consente di disaccoppiare la definizione formale del DTD del CML con l'effettiva implementazione grafica Java3D di ciascun componente sintattico e semantico del linguaggio di descrizione del mondo. In tal modo, ogni

variazione al DTD impatterebbe solamente sul Parser, ma non sull'implementazione grafica all'interno dello Scene Builder, e conversamente una diversa implementazione grafica di un elemento riguarderebbe una correzione all'interno di Scene Builder, ma nessuna modifica a livello di parsing del DTD:

Il processo di montaggio può avvenire solo a valle della fase di login, avendo cioè ottenuto l'accesso via rete al CML da interpretare, ed al database contenente le geometrie a cui la descrizione CML fa riferimento. Il database può contenere le geometrie dei componenti in uno qualsiasi dei formati grafici interpretabili dalle classi loader disponibili allo Scene Builder. Parte degli oggetti possono dunque essere modellate in 3DS, altre in VRML, altre ancora in X3D. Tutte andranno a confluire, successivamente alla fase di montaggio, nell'unico Grafo di Scena calcolato dallo Scene Builder.

La prima fase del montaggio si occupa di individuare e renderizzare il *contenitore ambientale* che rappresenta la parte, vale a dire la geometria dell'ambiente che racchiude tutti gli altri oggetti e definisce lo spazio di navigazione degli utenti per quanto riguarda la parte rappresentata dal CML considerato. Si tratta solitamente di una geometria rappresentante una stanza, o un luogo chiuso che costituisce l'ambito della collaborazione. Solo in una seconda fase si passa al montaggio degli oggetti statici distribuiti all'interno del contenitore ambientale, e successivamente al montaggio degli oggetti condivisi e delle geometrie che rappresentano gli avatar degli utenti.

Mentre il montaggio degli oggetti statici è un procedimento concettualmente del tutto identico alla renderizzazione del contenitore ambientale, il tracciamento degli oggetti condivisi (gli oggetti il cui stato deve essere replicato su tutte le istanze in rete dello spazio condiviso, e su cui gli utenti hanno possibilità di interazione) e degli avatar richiede, dopo un montaggio statico degli oggetti nella posizione di partenza specificata nel CML, l'aggiornamento allo stato condiviso corrente, fornito dallo strato di gestione dello stato condiviso, raggiungibile via rete grazie alla precedente fase di collegamento.

Poiché infatti le caratteristiche geometriche (posizione, orientamento e grandezza) e visive (colore, texture...) di avatars e oggetti condivisi possono variare in seguito a determinate interazioni collaborative tra utenti e oggetti dello spazio, è necessario trasmettere eventuali cambiamenti a tutti gli utenti presenti contemporaneamente nel mondo, in modo da garantire la coerenza dell'universo virtuale: se un corpo viene spostato o modificato, chiunque deve essere in grado di vedere la nuova posizione occupata dall'oggetto ed il suo nuovo aspetto.

A questo scopo, lo SceneBuilder mantiene localmente due tabelle di stato, una per gli avatar e una per gli oggetti condivisi.

Le informazioni prelevate dal Parser per il montaggio riguardano il percorso in rete dell'elemento geometrico (prelevato dal database), le coordinate spaziali di montaggio dell'oggetto all'interno della parte, e l'aspetto visivo (ad esempio l'URL della texture da applicare all'oggetto), nonché la classe Java da agganciare eventualmente come comportamento all'oggetto.

Il montaggio dinamico di ciascun oggetto all'interno del grafo di scena complessivo avviene creando un *BranchGroup* capostipite per ogni nodo da montare specificato a livello di CML. Si noti che il *BranchGroup* è l'unico nodo di un grafo di scena che può essere agganciato al grafo globale a runtime. Questo consente, tra l'altro, di renderizzare la scena attraverso un parsing progressivo del CML ed un parallelo rendering geometrico progressivo: l'utente assiste dunque al progressivo costruirsi della scena a video, mentre le informazioni vengono analizzate, e le geometrie vengono disegnate e posizionate nella rappresentazione tridimensionale.

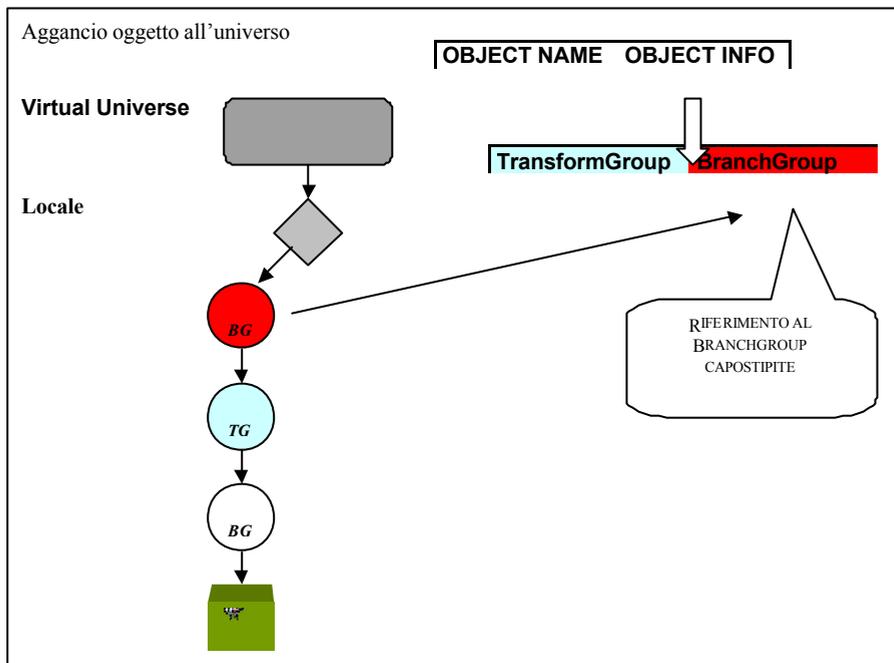
Il posizionamento effettivo dell'oggetto viene ottenuto calcolando una matrice di trasformazione  $4 \times 4$  in coordinate omogenee<sup>19</sup>, che esprime, rispetto al sistema di riferimento globale, la posizione dell'oggetto in termini di rotazione, traslazione e scalatura applicate.

La matrice di trasformazione viene inglobata, sotto forma di oggetto *Transform3D*, in un nodo *TransformGroup*, che può essere agganciato immediatamente al di sotto del *BranchGroup* per determinare il posizionamento della geometria. L'oggetto *TransformGroup* riveste una particolare importanza nella rappresentazione dello stato dell'oggetto: ad esempio, se è possibile ruotare l'oggetto con il mouse, è il *TransformGroup* contenente la geometria dell'oggetto ad essere modificato in seguito allo spostamento del mouse.

---

<sup>19</sup> Per una trattazione completa delle matrici omogenee di trasformazione (quaternioni) ci si può riferire a Foley, Van Dam, *Computer Graphics Principles and Practice, 2nd Edition, Addison Wesley*

Per gli oggetti condivisi è necessario tenere traccia dei loro TransformGroup, per monitorare eventuali cambiamenti o per effettuarne. A tale scopo, lo SceneBuilder mantiene una tabella contenente nome dell'oggetto condiviso, a cui viene assegnato un identificativo univoco, ed un particolare campo, composto dal TransformGroup e da un riferimento al suo BranchGroup capostipite. La memorizzazione di un riferimento al BranchGroup capostipite nel vettore di stato degli oggetti condivisi è necessaria per la fase di smontaggio dinamico dell'oggetto, mentre il riferimento al TransformGroup nel vettore di stato degli oggetti consente di accedere rapidamente alle informazioni



**Fig. 5-2: Il BranchGroup generato dallo SceneBuilder dinamicamente, a fronte dell'interpretazione di un nodo CML. La struttura del vettore di stato degli oggetti dinamici consente la gestione dello stato e il successivo smontaggio dinamico, attraverso la memorizzazione di un riferimento al nodo di Trasformazione e al nodo di montaggio capostipite nel ramo del grafo di scena**

posizionali e modificarle per riflettere uno stato condiviso modificato o una interazione locale dell'utente. L'unione dei riferimenti ai nodi TG e BG all'interno della tabella dello stato degli oggetti forma una struttura dati che viene convenzionalmente denominata 3Dinfo.

### 5.2.2.3 Transito a nuove Parti del Mondo

L'organizzazione della struttura dell'ambiente tridimensionali prevede la divisione dell'interno "mondo" in più parti. Questa divisione è dettata sia dall'esigenza di avere un criterio di partizionamento dei contenuti, che consentisse di creare situazioni diverse di rappresentazione dell'informazione, a cui eventualmente legare diverse modalità di cooperazione, sia da una necessità puramente implementativa, per cui risulta più performante suddividere in parti più piccole gli scenari grafici da renderizzare e da mantenere sincronizzati dal punto di vista dello stato condiviso.

Ogni volta che un utente entra in una zona corrispondente ad una nuova parte è necessario eseguire una procedura simile a quella di log-in, per poter costruire le geometrie dei nuovi oggetti e degli avatar, e per aggiornarne lo stato. La parte che viene lasciata viene smontata e al suo posto viene montata quella nuova. Attraverso la gestione dello stato condiviso, deve essere notificato a tutte le istanze remote dello spazio condiviso, che un particolare utente lascia una parte per entrare in un'altra. Similmente, l'ingresso dell'utente nella nuova parte va notificato a tutte le altre istanze che sono collegate alla parte di destinazione. Lo Scene Builder gestisce queste procedure.

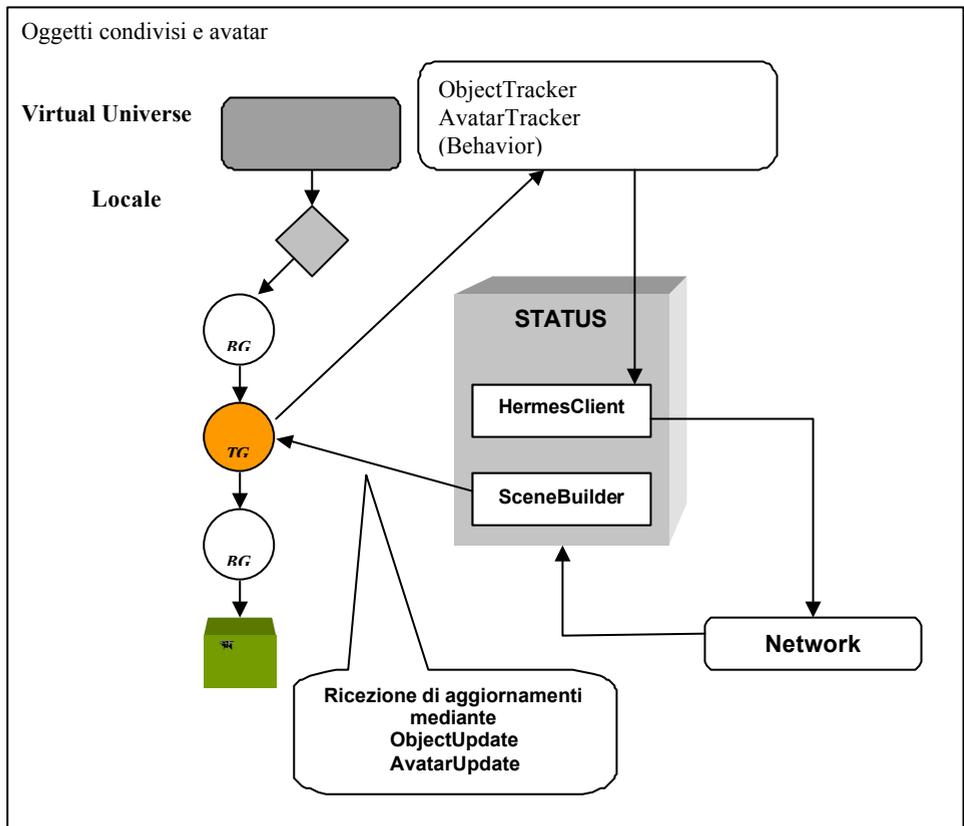
### 5.2.2.4 Aggiornamento di Oggetti Condivisi ed Avatar

Gli oggetti condivisi e gli avatar sono due categorie di nodi geometrici i cui attributi devono essere continuamente aggiornati all'interno della rappresentazione tridimensionale.

Gli oggetti condivisi rappresentano componenti del mondo con cui l'utente è in grado di interagire: l'interazione provoca una variazione di stato, modificando uno o più degli attributi dell'oggetto interessato (ad esempio la posizione, l'apparenza, il comportamento). L'effetto di questa interazione deve essere distribuito a tutte le istanze remote dello spazio condiviso, per mantenere la percezione di condivisione presso tutti i partecipanti.

Gli avatar, invece, rappresentano oggetti il cui scopo è rappresentare la posizione nello spazio di ciascuno dei partecipanti. Poiché i partecipanti si trovano in continua esplorazione, muovendosi nello spazio condiviso, è cruciale che questi movimenti vengano riprodotti il più fedelmente possibile in ciascuna delle istanze dello spazio condiviso.

Due sono gli aspetti che lo Scene Builder deve affrontare per realizzare l'effettiva condivisione di oggetti ed avatar. Da un lato, deve controllare ad una frequenza di campionamento prefissata, che le istanze locali degli oggetti condivisi non subiscano variazioni, o che l'utente non compia movimenti. Qualsiasi variazione di stato riscontrata al momento del campionamento deve tradursi in un calcolo della corrispondente matrice di trasformazione (sotto forma di oggetto Transform3D), e deve essere inviato via rete, al gestore dello stato condiviso, che si occupa di diramare questa informazione a tutte le altre istanze remote. Gli oggetti che si occupano del controllo per campionamento degli stati di oggetti e avatar all'interno del sistema sono



**Fig. 5-3: posizionamento di Tracker e Update per l'aggiornamento del grafo di scena rappresentante lo stato condiviso**

rispettivamente l'ObjectTracker e l'AvatarTracker. Essi sono implementati sotto forma di Behavior Java3D, e come tali sono soggetti alle regole di attivazione dei behavior legate ai bound di attivazione (v. oltre).

L'invio delle informazioni di aggiornamento dall'istanza locale al gestore dello stato condiviso avviene attraverso uno strato di comunicazione denominato HermesClient, progettato per comunicare con lo strato di rete del software di gestione dello stato condiviso.

Dall'altro lato, occorre che gli eventi di cambiamento di stato occorsi in istanze remote dello spazio condiviso siano comunicate all'istanza locale. Il cambiamento di stato di un oggetto o di un avatar può avvenire nel momento in cui l'istanza corrente è collegata, o può essere data da azioni che sono avvenute prima dell'ingresso dell'istanza corrente allo spazio condiviso. La raccolta dei pacchetti di aggiornamento di stato avviene,

attraverso la rete (mediante un'interfaccia denominata NetListener) da parte delle classi ObjectUpdate e AvatarUpdate.

### 5.2.2.4.1 Tracking

L'attività di tracking su oggetti e avatar viene attivata nei seguenti casi:

- La navigazione da parte dell'utente nell'istanza locale attraversa il boundary di attivazione programmato per l'attivazione del comportamento. Solitamente si tratta di una geometria sferica, di grandezza proporzionale al tipo di oggetto, centrata sull'oggetto medesimo.
- Al trascorrere di un quanto di tempo che rappresenta la frequenza di campionamento per la verifica di cambiamento dello stato locale. La dimensione del quanto di campionamento è un parametro regolabile in funzione delle prestazioni locali e della dimensione complessiva (in termini di numero di utenti ed estensione geometrica) cui l'applicazione deve far fronte. Una frequenza di campionamento maggiore richiede un maggiore impegno sul lato dell'istanza locale dal punto di vista computazionale.

Se dall'ultimo ciclo di tracking risultano modifiche alla matrice di trasformazione applicata, la nuova matrice di trasformazione viene spedita attraverso le opportune primitive di rete messe a disposizione da un apposito strato denominato Hermes.

Anziché spedire l'intera matrice a coordinate omogenee rappresentante la trasformazione, è necessario determinare la natura della trasformazione e spedire solo la partizione di matrice che è effettivamente variata, nell'ottica di mantenere basso il valore.

Si possono determinare fondamentalmente quattro casi<sup>20</sup>:

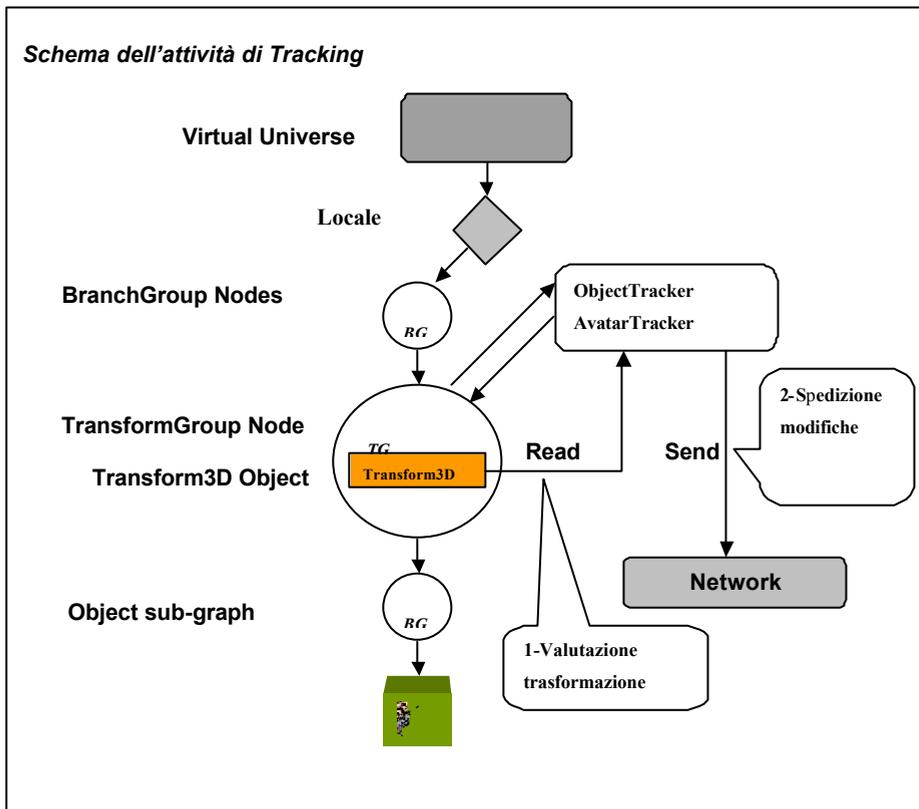
- Nel caso di traslazione semplice, è sufficiente inviare il vettore di tre elementi che descrive la direzione di spostamento della traslazione.
- Nel caso di una rotazione occorre inviare la sottomatrice 3x3 che descrive la rotazione secondo l'asse prescelto, o l'angolo theta e l'asse di rotazione

<sup>20</sup> Riportiamo i quaternioni di traslazione, rotazione su asse x e scalatura per riferimento. Le trasformazioni composte si ottengono per moltiplicazione in ordine inverso dei quaternioni base.

$$\begin{matrix}
 \begin{matrix}
 T \\
 \begin{matrix}
 ?1 & 0 & 0 & ?_x \\
 ?0 & 1 & 0 & ?_y \\
 ?0 & 0 & 1 & ?_z \\
 ?0 & 0 & 0 & 1
 \end{matrix}
 \end{matrix}
 &
 \begin{matrix}
 R_x \\
 \begin{matrix}
 ?1 & 0 & 0 & 0 \\
 ?0 & \cos? & ? \sin? & 0 \\
 ?0 & \sin? & \cos? & 0 \\
 ?0 & 0 & 0 & 1
 \end{matrix}
 \end{matrix}
 \\
 \\
 \begin{matrix}
 S_{xyz} \\
 \begin{matrix}
 ?_x & 0 & 0 & 0 \\
 ?_y & ?_y & 0 & 0 \\
 ?_z & 0 & ?_z & 0 \\
 ?_0 & 0 & 0 & 1
 \end{matrix}
 \end{matrix}
 \end{matrix}$$

- In una trasformazione di scala è sufficiente inviare i tre elementi di diagonale della matrice di trasformazione, recanti i fattori di scala lungo ciascuno dei tre assi.
- Nel caso di trasformazioni composte verranno spedite le componenti che hanno subito variazioni, ad esempio se si verifica una roto-traslazione, verranno passati il vettore di traslazione e la matrice di rotazione.

#### 5.2.2.4.2 Updating



**Fig. 5-4:** Schema generale dell'attività di tracking: alle condizioni di risveglio si valuta la matrice di trasformazione e si invia una notifica delle differenze alle istanze remote

Conversamente all'attività di tracking, l'attività di Update è preposta a rilevare gli aggiornamenti generati dalle istanze remote e comunicati all'istanza locale attraverso la rete. L'aggiornamento arriva all'istanza locale dello Scene Builder sotto forma di un oggetto serializzato. Questo oggetto contiene le seguenti informazioni:

- L'identificativo univoco dell'oggetto montato nel grafo di scena, che ha subito l'aggiornamento, o in alternativa l'indirizzo IP e la porta di connessione dell'utente il cui avatar ha subito un aggiornamento di stato
- La porzione di matrice di trasformazione da aggiornare sull'oggetto in questione

La trasformazione da porre sull'oggetto viene ottenuta ricostruendo la matrice di trasformazione completa a partire dalle informazioni parziali contenute nell'oggetto trasmesso.

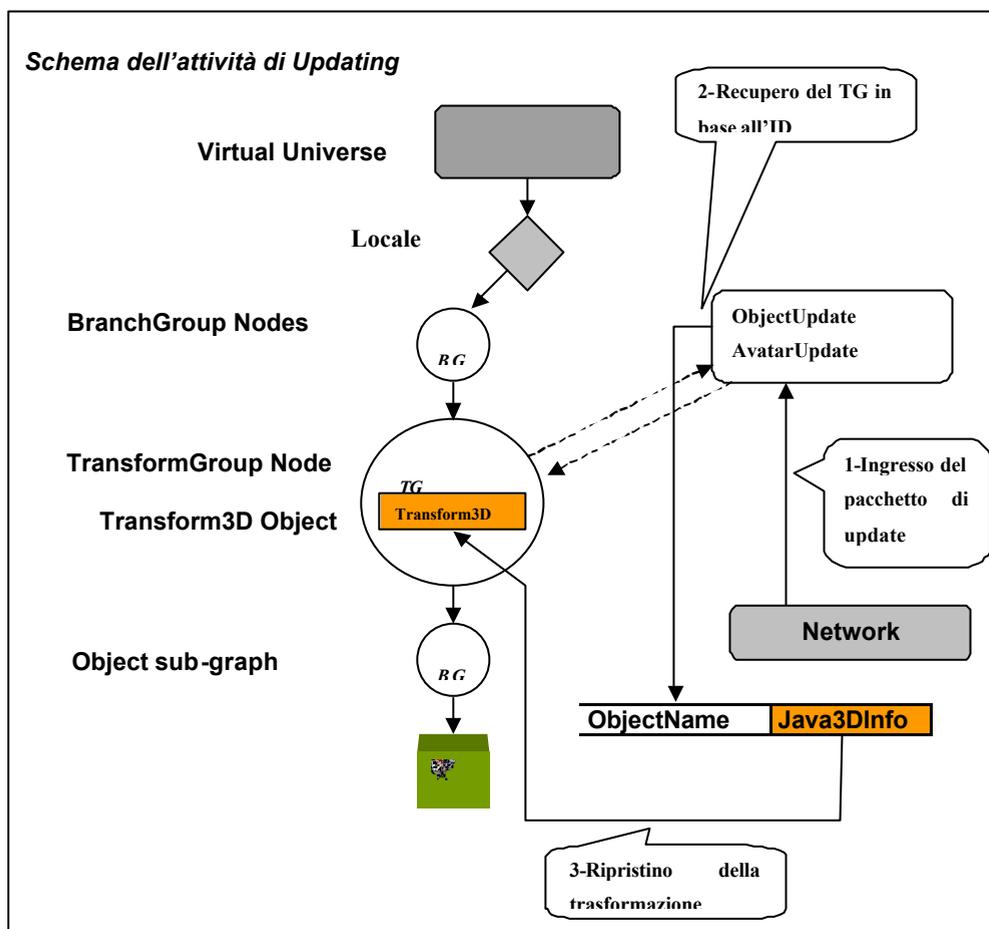


Fig. 5-5: schema di funzionamento della procedura di Update: il frammento di trasformazione viene ricostruito in una trasformazione completa. L'ID dell'oggetto viene usato per recuperare il riferimento al nodo transformgroup locale a cui applicare l'aggiornamento della trasformazione.

L'istanza locale contiene una tabella la cui chiave è l'ID dell'oggetto oppure l'IP dell'avatar che ha subito l'update. Il valore della tabella è la classe composta da TG e BG capostipite dell'oggetto considerato. Dal riferimento al TransformGroup prelevato in tabella è possibile applicare la nuova trasformazione recuperata dal frammento di aggiornamento arrivato via rete.

### 5.2.2.5 Associazione di comportamenti agli oggetti

Il concetto di comportamento (Behavior) è il secondo aspetto fondamentale offerto da Java 3D, che consente di realizzare in modo relativamente semplice concetti tipici dei sistemi collaborativi, vale a dire offrire diverse modalità di interazione tra utenti ed oggetti, mediante sistemi diversi di manipolazione e di risposta offerti dagli oggetti stessi.

Il concetto di Behavior è fornito in Java 3D attraverso la classe omonima: un behavior ha un criterio di attivazione (Wakeup Criterion), un *boundary* di attività, entro il quale il criterio di attivazione viene considerato, e fuori del quale il criterio viene ignorato, e una parte di processStimulus, dove viene descritto, mediante programmazione Java, come rispondere allo stimolo ricevuto.

L'aggancio di un Behavior sotto ad un BranchGroup consente di realizzare varie funzioni, come ad esempio rendere sensibile alla selezione o al trascinamento del mouse l'oggetto, eventualmente mappando i movimenti del mouse ad una matrice di trasformazione presente nel medesimo BranchGroup. Poiché in generale è possibile descrivere il comportamento di un oggetto in modo programmatico, posto che la classe Java eredita direttamente dalla classe Java3D Behavior, si profila un modo per consentire la scrittura di generici comportamenti di interazione con l'utente per generici oggetti, avendo a disposizione la potenza e la flessibilità di un intero linguaggio.

Mediante il meccanismo Java della *reflection*, è possibile caricare a partire da un URI specificato un file .class, ed utilizzarlo a runtime. Le API Reflection consentono di esplorare la classe caricata in memoria, e determinarne metodi e costruttori. Fornendo a priori determinati metodi di interfacciamento con i meccanismi di manipolazione del grafo di scena operati all'interno dello Scene Builder, è possibile dunque caricare a runtime (e montare dinamicamente, seguendo le istruzioni del CML) porzioni di programma che descrivono modalità generiche di interazione dell'oggetto con altri oggetti o con l'utente.

Una particolare porzione dello Scene Builder si occupa, attraverso la reflection, di caricare le classi e di montarle all'interno di un nuovo BranchGroup. Questo BranchGroup viene poi attaccato sotto il BranchGroup capostipite dell'oggetto cui il comportamento deve collegarsi. Montando il BranchGroup contenente il comportamento sotto il generale TransformGroup preposto all'animazione dell'oggetto, è possibile collegare gli stimoli raccolti del Behavior e trasformarli in particolari movimenti dell'oggetto stesso.

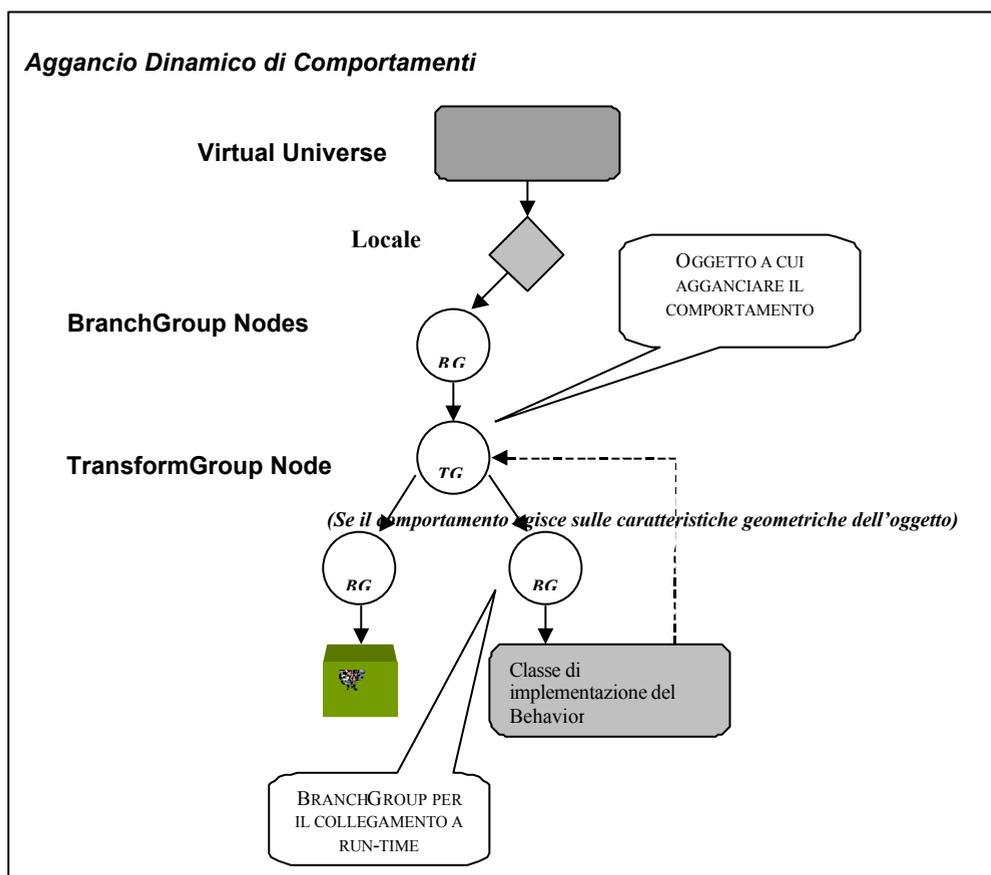
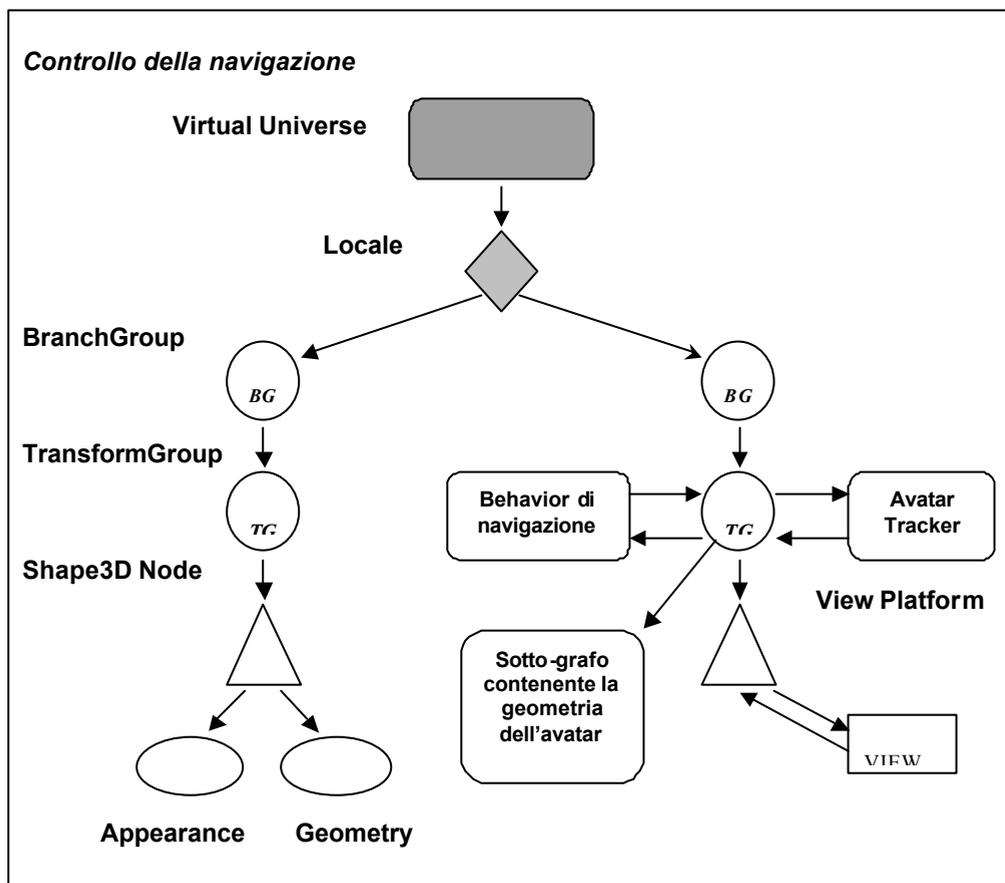


Fig. 5-6: L'aggancio dinamico di comportamenti avviene attraverso l'utilizzo delle API Reflection, e la creazione di un nuovo BranchGroup, innestato sotto il BranchGroup capostipite dell'oggetto.

### 5.2.2.6 Gestione della navigazione e dei punti di vista



**Fig. 5-7:** la gestione della navigazione avviene lavorando sul branch di visualizzazione del grafo di scena. Per gestire la navigazione occorre un comportamento di rilevamento dei comandi locali di spostamento, ed un comportamento di tracking per l'aggiornamento delle istanze remote.

Una volta costruito, mediante il ramo dei contenuti grafo di scena, l'universo tridimensionale, l'altro ramo – di visualizzazione, va utilizzato per gestire due aspetti fondamentali dell'interazione con l'utente:

- lo spostamento dell'utente all'interno della scena (navigazione), e la relativa raccolta dei suoi dati di spostamento, ai fini di comunicarli alle istanze remote dello Scene Builder
- la gestione di punti di vista diversi, non necessariamente coincidenti con la posizione di navigazione dell'utente

Il movimento attraverso la scena viene implementato attraverso l'aggancio di due Behavior ad un TransformGroup sotto cui montare la piattaforma di visualizzazione Java3D: il Behavior Navigazionale ha il compito di rimanere in ascolto per il verificarsi di eventi di tastiera che comandano lo spostamento dell'utente (avanti, indietro, etc.); il secondo Behavior è invece il tracking preposto al rilevamento dello spostamento e all'invio di questa informazione alle istanze remote dello Scene Builder.

Per gestire diversi punti di vista (ad esempio posizionare una telecamera alle spalle dell'utente, in modo che l'utente possa vedere non una visualizzazione puramente soggettiva, ma una rappresentazione del mondo in cui è possibile vedere anche la geometria che rappresenta il proprio avatar), al TransformGroup del ramo di visualizzazione è agganciata anche la geometria che descrive l'avatar dell'utente locale. In questo modo, è possibile cambiare la posizione della Viewing Platform, includendo anche la geometria dell'avatar locale. Il punto da sottolineare è che la navigazione con il Behavior di navigazione impone uno spostamento della geometria dell'avatar e della relativa viewing platform in parallelo.

### **5.2.3 Conclusioni sull'uso di Java 3D**

L'utilizzo di Java 3D ha messo a disposizione della nostra architettura collaborativa due potenti concetti: il grafo di scena orientato agli oggetti, e i behavior con possibilità di reflection. Mediante questi due strumenti diventa possibile modellare un'applicazione in grado di aggiornare lo spazio condiviso tridimensionale, e il cui comportamento è modellabile attraverso un linguaggio di programmazione.

Se da una parte l'utilizzo di uno strato di programmazione ad oggetti consente di raggiungere un alto livello di astrazione e programmabilità, dall'altra occorre pagare un prezzo relativo in termini di pura prestazionalità grafica. Alcune scelte tipiche nella ricerca di prestazioni in fase di real-time rendering, quali la politica di frustum culling, l'algoritmo di clipping e di identificazione delle superfici nascoste, la scelta del livello di dettaglio delle mesh poligonali che compongono la scena, e via dicendo, sono tolte dalle mani dello sviluppatore e completamente delegate al motore interno Java 3D.

Resta tuttavia da valutare quanto l'effettiva possibilità di manovrare queste leve dia un margine prestazionale sensibile. Per poterlo determinare, abbiamo implementato alcune politiche di rendering real-time direttamente in OpenGL, e misurato il guadagno prestazionale ottenibile con queste tecniche rispetto alle politiche di rendering *built-in* del motore Java3D.

### **5.3 Tecniche di Rendering Prestazionale**

Alcune tecniche proprie del rendering in *real-time* di immagini tridimensionali possono venire in aiuto per aumentare le prestazioni della rappresentazione offerta all'utente nell'ambito dell'istanza locale. Tra queste tecniche è possibile citare una maggiore gestione del *clipping*, della *collision detection*, e una più performante gestione del

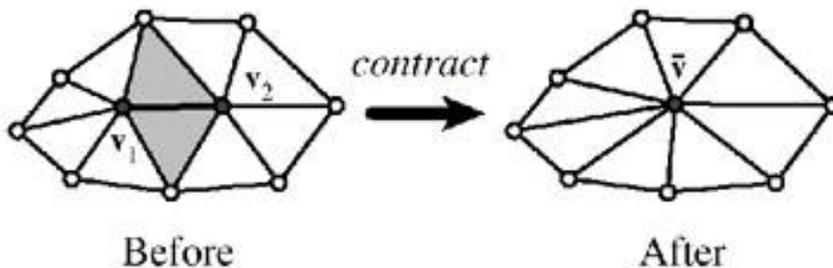
dettaglio di visualizzazione. Queste tecniche possono essere esplorate, tuttavia, al di fuori dell'ambito di Java3D, poiché Java3D segue una propria politica e ordine di rendering dei poligoni montati nel proprio grafo di scena.

La tecnica di gestione del dettaglio di visualizzazione, in particolare, risulta particolarmente interessante nell'ottica di utilizzo di sistemi collaborativi tridimensionali per l'esplorazione di gruppo di ricostruzioni artistiche ed archeologiche, dove si vuole offrire un elevato livello di dettaglio per scopi scientifici quando si esamina in prossimità un sito od un oggetto, e tuttavia mantenere elevate prestazioni di navigazione durante il movimento.

La soluzione al dilemma è costituita dalla gestione, da parte del motore di renderizzazione grafica, di maggiori livelli di dettaglio (LOD): in altre parole, il renderer deve utilizzare un elevato numero di poligoni quando la distanza tra oggetto e osservatore è piccola, e l'osservatore è in grado di apprezzare i dettagli della scena o dell'oggetto vicino a cui si trova. Viceversa, al crescere della distanza, il numero di poligoni necessario a costruire la medesima geometria può decrescere, perché al rimpicciolire delle dimensioni dell'oggetto osservato diminuisce la possibilità di notare i dettagli.

Questa tecnica, denominata MultiLOD (Multiple Level of Detail) è stata sperimentalmente implementata in un prototipo di Scene Builder che opera direttamente utilizzando OpenGL, per valutarne le prestazioni e conseguentemente l'interesse in un suo innesto in un futuro Scene Builder ad alte prestazioni.

La tecnica di *LODding* utilizzata segue la strategia di collassamento per vertici<sup>21</sup>.



**Fig 5-8: Gestione del livello di dettaglio per collassamento dei vertici**

L'oggetto in esame, per costruzione, è realizzato con un certo numero di poligoni, in particolare di triangoli: questa scelta è dovuta ad esigenze di complanarità: tre punti, infatti, giacciono sicuramente sullo stesso piano ed il poligono risultante dall'unione di essi non genera problemi per quanto concerne il calcolo delle normali e di eventuali effetti di riflessione della luce.

<sup>21</sup> R. Borgo, P. Cignoni, R. Scopigno - An Easy-to-use Visualization System for Huge Cultural Heritage Meshes in *proceedings ACM VAST 2001, Atene, Dicembre 2001*

Partendo da questa struttura dati è possibile individuare per ciascun vertice i punti con cui esso risulta in contatto: si ha quindi una lista ordinata dei vertici e dei triangoli che compongono l'oggetto in esame ed è possibile ricavare da questa delle informazioni più dettagliate sulla struttura che si sta studiando.

Il dato utilizzato nel collassamento per vertici è la distanza che separa gli uni dagli altri (si considerano per questo scopo solamente le distanze tra vertici adiacenti ed in contatto tra loro). Seguendo un procedimento ricorsivo si esplorano tutti i vertici fino a trovare i due che tra loro hanno distanza minore: questi due elementi sono i candidati per il collassamento. Valutando la distanza che ciascuno di questi ha a sua volta con i propri vicini si determina l'effettivo elemento da collassare: tale vertice viene portato sul suo adiacente (precedentemente candidato al collassamento) in modo da ridurre il numero di vertici della figura. Passo successivo è fare in modo che tutti i vertici adiacenti a quello collassato abbiano ora come adiacente il vertice su cui è stato collassato l'elemento tolto dall'algoritmo di riduzione.

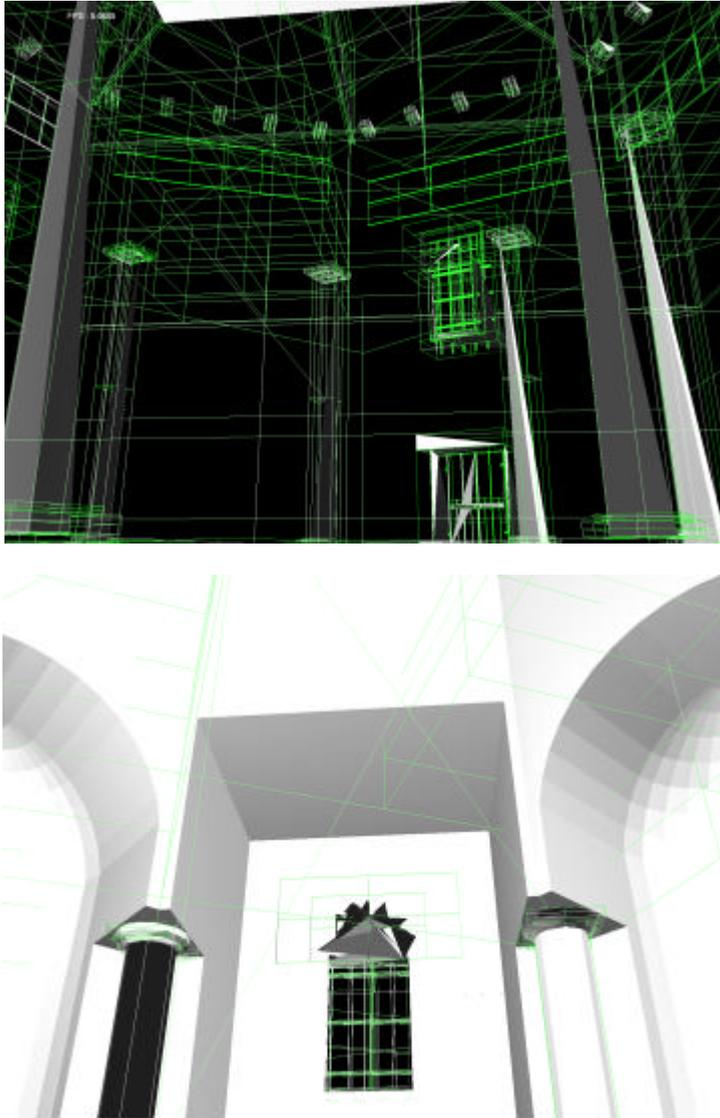
La nuova lista di vertici e di triangoli viene completata e trasferita alla parte del motore grafico designata alla renderizzazione. Bisogna notare che questo procedimento (calcolo dei candidati, collassamento, disegno) andrebbe ripetuto ogni volta che l'oggetto si sposta nello spazio dal momento che il numero di poligoni che si sceglie di usare è derivato dalla distanza che separa l'osservatore dallo oggetto in esame (più un oggetto è distante, meno poligoni si utilizzano in modo da ridurre i calcoli che la CPU o la scheda grafica devono eseguire per il rendering). Per ovviare a questo oneroso processo è possibile applicare l'algoritmo di riduzione tutto in una volta e conservare in memoria una sorta di "storia" delle modifiche che sono state applicate all'oggetto per ogni data distanza: in questo modo si allungano i tempi di inizializzazione del modello, ma si riducono i tempi richiesti nel corso dell'esplorazione del mondo dal momento che la CPU o la scheda video non devono più applicare il multiLOD, limitandosi a caricare le diverse strutture dati dalla memoria dove già si trovano ad inizializzazione terminata.

I risultati ottenuti mediante questa tecnica sono significativi: si registrano aumenti di prestazioni di circa il 30% rispetto al motore Java3D privo di una gestione del dettaglio.

A riprova dell'efficacia di questa tecnica, si è creato un modello architettonico di elevato dettaglio, riprodotto la Basilica di Santa Tecla in Milano. La navigazione collaborativa all'interno del motore Scene Builder Java3D è risultata impraticabile a causa dell'elevatissimo numero di poligoni, necessari a rendere con precisione ogni dettaglio.

L'utilizzo invece di un microkernel di visualizzazione, in grado di elaborare gli indexed face set VRML del modello e di renderizzarli in OpenGL attraverso una politica di multiLODding ha consentito di esplorare e navigare il modello.

L'implementazione nel microkernel OpenGL realizzato è dotata anche della possibilità di regolare un fattore di peso di collassamento dei poligoni in relazione alla distanza dell'osservatore (lunghezza del frustum di visualizzazione). In questo modo è stato possibile osservare empiricamente l'effetto del collassamento in termini di precisione di visualizzazione (misurate empiricamente con osservazioni qualitative) e prestazioni di navigazione (misurate attraverso gli FPS, frame per second nel tracciamento della scena).



**Fig 5-9:** uno scorcio del modello ad alta risoluzione “La Basilica di Santa Tecla”, utilizzato per valutare la fattibilità di esplorazioni collaborative su modelli archeologici ad alta precisione. Nella figura superiore, il selezionatore di poligoni utilizza un coefficiente di riduzione elevato. Nella visualizzazione delle bounding box dei sottogruppi poligonali componenti il modello, solo pochi poligoni sono effettivamente selezionati per il rendering. Sotto, il selezionatore poligonale utilizza un coefficiente di selezione minore, e l’osservatore si è avvicinato per osservare il dettaglio. Viene tracciato un maggiore numero di poligoni, ma il frame rate di tracciamento diminuisce



**Fig. 5-10:** effetto del collassamento per vertici sulla precisione del modello, in un esperimento eseguito sulla geometria rappresentante la “chiesa ideale” disegnata da Leonardo da Vinci. Ad elevata distanza, il selettore poligonale collassa la mesh poligonale in modo molto pronunciato: tuttavia, la distanza è tale che questo è scarsamente visibile, e le prestazioni di rendering raggiungono i 53 FPS (53 ritracciamenti di scena al secondo, che rendono la navigazione estremamente fluida). Avvicinandosi, si nota come alcuni poligoni risultino ancora collassati (11 FPS). A distanza ravvicinata, la mesh è presentata al massimo dettaglio, ma la velocità di navigazione è ridotta (9 FPS). Tuttavia, a questo livello di osservazione, la fluidità di movimento richiesta dall’osservatore è solitamente minore (i dettagli vengono osservati stazionariamente).

## 5.4 Conclusioni

I concetti di *grafo di scena* e di *behavior* sono di estrema importanza per l'implementazione elegante di tutte le funzionalità necessarie al mantenimento dello stato condiviso e della sua rappresentazione tridimensionale. Grazie a questi concetti, è stato possibile elaborare delle politiche di aggiornamento della scena e di interazione con l'utente che risultano semplici concettualmente e relativamente semplici implementativamente.

Le prestazioni ottenute sono soddisfacenti, ma non sufficienti per reggere la visualizzazione in tempo reale di un gran numero di spostamenti, o per tracciare geometrie ad elevata precisione e complessità. Per risolvere questi aspetti, occorre scendere di livello nell'implementazione, e controllare finemente il modo con cui il rendering delle scene e il controllo dell'interazione viene gestito a basso livello.

Il futuro per quanto riguarda lo strato ad implementazione grafica sembra prendere una via più complessa: richiedere cioè l'implementazione di un *renderer* real time ad hoc, scritto direttamente in linguaggio OpenGL, in cui sia possibile effettuare direttamente varie scelte di ottimizzazione delle prestazioni. Tuttavia, occorrerà organizzare la struttura interna di questo *renderer* in modo da riprodurre i concetti preziosi offerti da Java3D: progettare, sul calco del grafo di scena attualmente utilizzato, un nuovo grafo di scena, eventualmente potenziato, ed implementare un meccanismo che consenta la modellazione di *behavior*.

Un ulteriore fattore di variabilità potrebbe essere dato dalla finalizzazione, dopo lungo tempo, dello standard X3D<sup>22</sup>. Questo formato, basato su XML, consentirebbe di espandere facilmente linguaggi standard utilizzando nodi e semantiche ad hoc, ma specificate in modo standard. In questo scenario, un motore di rendering ottimizzato per il supporto della collaborazione in ambito tridimensionale, e scritto espressamente a questo scopo, potrebbe utilmente far uso di una estensione dello standard orientata alla descrizione della visualizzazione di elementi ed eventi in contesti collaborativi. Questo costituirebbe un ulteriore punto a favore della scelta di percorrere fino in fondo la strada della reimplementazione del motore grafico.

---

<sup>22</sup> [www.x3d.org](http://www.x3d.org)

## 6 La collaborazione assistita

### 6.1 *Necessità di collaborazione assistita*

Nel tentativo di comprendere meglio la natura della collaborazione tra utenti all'interno di ambienti tridimensionali condivisi, sono state sviluppate varie applicazioni, ciascuna volta a cercare di raccogliere osservazioni ed esperienze circa determinati aspetti della progettazione e della realizzazione di questo tipo di applicazioni.

Una delle prime applicazioni realizzate, Leonardo Virtuale, più volte citata nei capitoli precedenti come esempio su cui appoggiare la discussione concettuale di vari aspetti di questa tipologia di sistemi, ha potuto servire in modo continuativo gli utenti del sito web del Museo della Scienza e della Tecnica di Milano. Durante il tempo di servizio dell'applicazione, è stato scelto un periodo di osservazione, durante il quale si sono registrati in un file di log numerosi parametri di utilizzo dell'applicazione<sup>23</sup>.

I parametri osservati sono stati svariati, tra cui:

- il numero di tentativi di accesso al mondo collaborativo da parte dello stesso utente
- la frequenza di utilizzo delle funzionalità offerte, suddivise tra la possibilità di esplorare il mondo tridimensionale navigando, la possibilità di interagire con gli oggetti, di visitare le pagine web collegate alla struttura tridimensionale, di parlare con altri utenti
- il tempo trascorso all'interno dell'applicazione in relazione al verificarsi di alcune situazioni quali la presenza contemporanea di un certo numero di altri utenti; la presenza di una guida qualificata; il verificarsi di dubbi o problemi tecnici
- il volume di ingressi al sistema in corrispondenza alla disponibilità di visite guidate da parte di utenti esperti della tematica esposta nel museo collaborativo

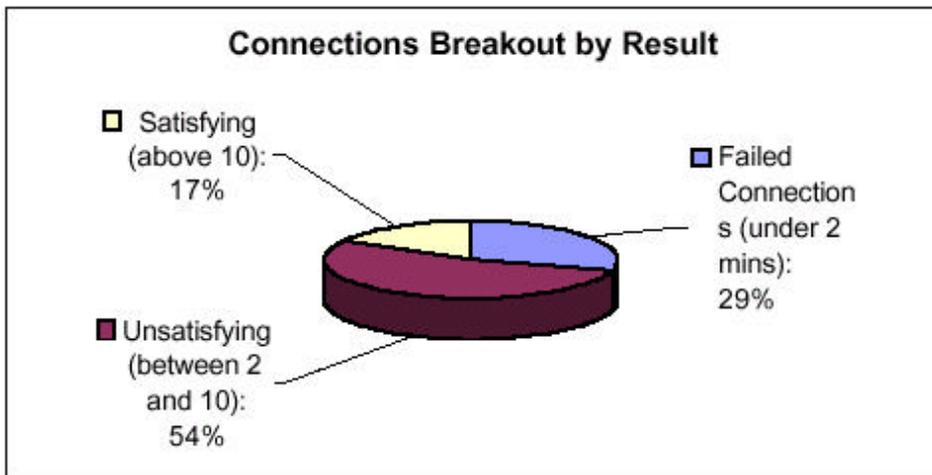
Riassumendo i dati raccolti, è stato possibile comprendere come la collaborazione in un ambiente tridimensionale risulta in realtà difficile per un utente che non abbia una cultura tecnica o una nozione esatta dello scopo per cui il sistema è stato realizzato. Sebbene il sistema effettivamente faciliti la collaborazione tra più persone, come comprovato dall'osservazione di numerosi casi in cui gli utenti si sono effettivamente incontrati e scambiati opinioni e commenti, navigando all'interno dell'informazione, molto spesso l'incertezza su come effettivamente si dovesse utilizzare lo strumento, o l'insorgere di problemi puramente tecnici (mancanza di memoria RAM, erronea configurazione del software di navigazione, etc.) hanno impedito di realizzare un'effettiva collaborazione.

---

<sup>23</sup> Barbieri T., Paolini P. – “CooperativeVisits to WWW Museum Sites a Year Later: Evaluating the effect”, in *ProceedingMuseums&Web2000, Minneapolis (USA), April 2000*

Un aspetto particolarmente importante è la mancanza di confidenza con le modalità di funzionamento ed in generale con la metafora di collaborazione rappresentata dall'intero sistema collaborativo tridimensionale. Si è verificato che quando gli utenti erano presenti nel sistema in un certo numero, il reciproco scambio di informazioni e il reciproco aiuto ha portato a una più rapida comprensione dell'utilizzo dell'ambiente e dell'esplorazione dei contenuti. Quando invece gli utenti si sono trovati ad esplorare l'ambiente in modo solitario, senza che nessun meccanismo di collaborazione si fosse effettivamente innescato, i problemi tecnici e di esplorazione sono risultati insormontabili, oppure l'interesse generato nell'utente è risultato essere piuttosto limitato.

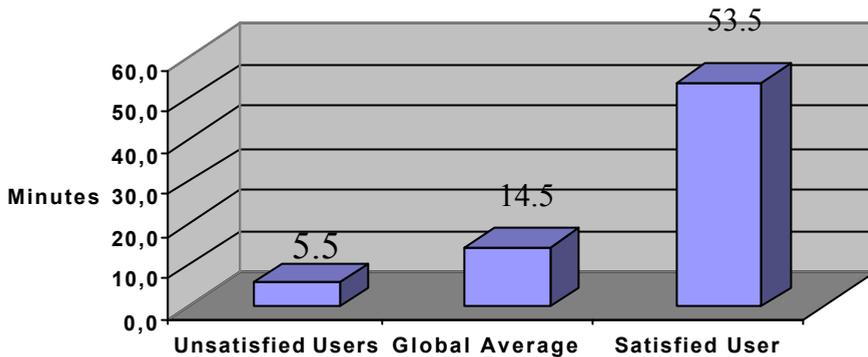
I dati raccolti hanno confermato che, laddove ci sia una presenza di altri utenti o di un utente esperto (tecnicamente o nei contenuti) disponibili alla collaborazione, l'esperienza collaborativa tridimensionale risulta fattibile, proficua ed interessante. Laddove invece non ci sia un numero sufficiente di altri utenti con cui collaborare, i tempi di connessione sono minimi, ad indicare uno scarso interesse per l'applicazione, o una non comprensione *tout court* dei fini dell'applicazione.



**Fig. 6-1: Suddivisione delle connessioni in base al tempo di permanenza nel sistema. Nelle connessioni soddisfacenti il tempo di permanenza del sistema supera la mezz'ora.**

Nell'analisi dei tempi di permanenza delle connessioni, risulta che la percentuale di connessioni che supera i dieci minuti è piccola (17%), tuttavia la maggior parte di questa categoria di connessioni è avvenuta nei momenti di maggior popolazione dell'ambiente, o quando era disponibile un altro utente (anch'esso incarnato da avatar),

in grado di guidare gli utenti lungo i contenuti e all'uso del sistema. In questi casi il tempo medio di queste connessioni ha superato la mezz'ora.



**Fig 6-2: Tempi medi di permanenza in un sistema collaborativo: gli utenti che innescano un meccanismo di collaborazione rimangono per lungo tempo all'interno del sistema.**

Se ne deduce quindi un concetto molto semplice, e cioè che un sistema collaborativo è efficace solo qualora esso offra effettivamente una attività collaborativa. Ambienti tridimensionali collaborativi *desolati*, cioè privi di utenti, non hanno la capacità di trattenere al proprio interno un utente che vi entra per l'esplorazione.

Poiché tuttavia non è sempre possibile garantire un numero sufficiente di utenti reali all'interno del sistema, oppure la disponibilità di un utente guida che effettivamente rimanga all'interno del sistema, occorre pensare dei metodi alternativi per fare comunque in modo che il sistema, in ogni momento, offra un seppur minimo grado di collaboratività, in modo da consentire il progressivo popolamento con utenti reali, e parallelamente le possibilità di collaborazione offerte dall'ambiente.

Una soluzione a questo problema è la creazione di un meccanismo di *collaborazione assistita*: sotto questo termine indichiamo una serie di accorgimenti che siano volti a creare l'illusione della presenza di altri utenti con cui collaborare, senza che essi vi siano effettivamente. Questo risultato si può ottenere simulando altri utenti mediante l'attivazione di avatar automatici, con diverse funzionalità all'interno del mondo. L'intento di questi avatar pilotati dal sistema è quello di mantenere un livello minimo di popolazione, necessario a 'catalizzare' la presenza di utenti reali all'interno del mondo: in questo modo, offrendo un livello minimo iniziale di collaborazione, si conta di attirare e mantenere in posizione utenti reali, che possano offrire collaborazione effettiva agli utenti che entrano nel sistema successivamente.

Le funzioni che la collaborazione assistita offre all'interno dello spazio condiviso possono essere le seguenti:

- assistenza di base agli utenti, ai fini di addestrarli al corretto uso delle funzionalità di collaborazione offerte dal sistema, qualora queste non siano immediatamente comprensibili od evidenti
- guida ai contenuti in modo automatizzato od intelligente, creando utenti virtuali in grado di dirigere gli utenti reali all'esplorazione del mondo e dei suoi contenuti
- servizi di consulenza o di ricerca all'interno dei contenuti
- semplice funzione di presenza o popolazione, per limitare l'impressione di desolazione, o per indurre certi comportamenti, attraverso dimostrazione diretta, negli utenti reali.

Queste funzioni possono essere realizzate attraverso la rappresentazione di avatar legati non ad un utente reale, ma a degli algoritmi di animazione e controllo. Questi particolari avatar automatici vengono detti Agenti, quando le loro funzionalità implicano un'interazione diretta con un utente reale, e Puppet (pupazzi), quando invece non hanno alcune funzionalità di interazione, ma eseguono, indipendentemente dalla presenza di utenti reali o meno, delle azioni preprogrammate ripetute all'infinito e non modificabili dalle interazioni con altri utenti.

Elaboriamo ulteriormente nel seguito le caratteristiche qui elencate della collaborazione assistita.

### **6.1.1 Assistenza di base**

Le funzionalità di assistenza di base sono rivolte a determinare quando un utente si trova in situazioni di difficoltà e necessita di un aiuto o di un suggerimento. Il sistema di assistenza deve poter fornire le seguenti funzioni:

- *Stall Detection*: consiste nella possibilità di rilevare situazioni in cui l'utente si trova in stallo, non è cioè più in grado di pilotare il proprio avatar all'interno del mondo tridimensionale, perché si è perso, non conosce i controlli di navigazione, o non capisce che deve inviare un comando per spostarsi. Una semplice possibilità di intervento potrebbe essere quella di offrire aiuto utilizzando la chat e riportare un breve riepilogo dei comandi per il movimento, oppure quella di proiettare un avatar guida dentro la visuale corrente dell'utente, oppure ancora di far apparire una pagina di aiuto, o riportare, dietro consenso dell'utente, l'avatar in una posizione predefinita.
- *Controllo dei dialoghi*: possono essere resi disponibili alcuni comandi all'interno dell'area di conversazione che richiamano un avatar automatico di guida (Agente) all'interno della visuale dell'utente. L'agente potrebbe rispondere a domande specifiche su come fare per muoversi o per volare negli ambienti, oppure iniziare una conversazione (registrata o simulata attraverso la gestione di un sistema esperto) circa i contenuti dell'applicazione che l'utente sta navigando.
- *Neighbourhood activation*: L'agente potrebbe essere in grado di analizzare e reagire ad eventi combinati come la prossimità fisica dell'avatar dell'utente ad

un oggetto condiviso, modificabile, o associato ad un link del mondo. Se l'utente si avvicina ad un oggetto animabile o di cui è possibile sperimentare l'utilizzo modificandone lo stato, l'agente potrebbe attivare una sequenza automatica di movimento in cui mostra all'utente il proprio avatar che interagisce con quell'oggetto; in tal modo l'utente può essere invogliato a intraprendere la stessa azione, o comprendere che è possibile interagire con l'oggetto in quel modo, se la natura dell'interazione non è immediatamente intuitiva. In questo modo si aiuta l'utente non mediante una comunicazione diretta, ma per dimostrazione e imitazione: uno dei pattern comportamentali più comuni durante una sessione collaborativa reale. Tale sequenza automatica scatta in risposta ad eventi di prossimità provocati dall'utente o dalla mancanza di una esplicita interazione dell'utente con gli oggetti mobili.

La presenza di meccanismi di assistenza dunque, ruota attorno ad una comunicazione diretta tra agenti e utenti, e ad una reciproca osservazione: l'agente controlla lo stato dell'utente, e l'utente, osservando le azioni e le reazioni dell'agente, può comprendere meglio la natura del sistema in cui è immerso. In particolare, la comunicazione diretta tra agente e utente permette all'utente di richiedere aiuto all'agente come se questi fosse un altro utente dello spazio condiviso, e l'osservazione dell'agente da parte dell'utente può risultare didattica, egli può cioè apprendere l'esistenza di alcune modalità di utilizzo dell'ambiente 3D semplicemente osservando l'agente che ne mostra degli esempi.

Per citare un esempio, in molti ambienti virtuali esiste la possibilità di eseguire un'esplorazione "volando", non essendo cioè legati a camminare seguendo la geometria fisica dell'ambiente. Pochi tra i navigatori comprendono l'esistenza di questa possibilità, fino a che non osservano altri utenti navigare nell'ambiente usando questa modalità. Quando non ci sono altri utenti da osservare, la presenza di agenti che si spostano mediante la modalità di volo comunica direttamente questa possibilità all'utente che non ne è al corrente. Il passaggio dell'informazione avviene per dimostrazione: l'agente non spiega all'utente come si fa a volare, ma dimostra direttamente che esiste questa possibilità. Da questo momento l'utente può cercare attivamente il modo con cui eseguire questa azione (solitamente trovandola autonomamente). Il punto in questo caso non verte tanto sull'informare *come* eseguire una certa azione, ma semplicemente sull'esistenza della sua possibilità. Si tratta di un aspetto non banale all'interno di mondi tridimensionali condivisi: infatti spesso è possibile progettare modalità di accesso che non hanno un riscontro diretto con il reale, ma sono possibili solamente data la natura virtuale della rappresentazione. Tuttavia poiché solitamente gli ambienti rappresentano ambienti reali, la possibilità di azioni che solitamente non rientrano nella sfera dell'esperienza reale non viene percepita da un utente non esperto.

### **6.1.2 Guida Applicativa**

L'assistenza collaborativa nei termini di "guida applicativa" risulta utile ogniqualvolta il fine della collaborazione sia l'esplorazione in gruppo di informazioni, ad esempio culturali. In questi contesti, solitamente l'esplorazione in gruppo consente di individuare contenuti non direttamente ricercati, oppure di approfondire meglio alcuni aspetti dell'esplorazione. In particolare, la disponibilità di un utente esperto nell'applicazione e nei contenuti da essa proposti, dà la possibilità di porre domande specifiche e di navigare con maggiore approfondimento i contenuti dell'applicazione stessa, in tutta analogia con una guida museale.

Qualora non fosse sempre possibile una guida reale per vestire i panni di un avatar-guida all'interno di un'applicazione collaborativa, un'alternativa consiste nell'istruire un agente virtuale a condurre gli utenti lungo percorsi preventivamente registrati. L'agente può esporre informazioni riguardanti le tappe della visita e poi attendere che l'utente esprima il desiderio di passare alla tappa successiva fino al termine del percorso.

Considerando poi l'applicazione tridimensionale come una struttura di accesso collaborativa a contenuti eventualmente organizzati anche in un sito bidimensionale, l'agente si pone come guida interattiva e collaborativa alla ricerca di determinate porzioni del tradizionale sito web: si tratta di una modalità di esplorazione dei contenuti totalmente nuova, e diversa dal classico approccio a "motore di ricerca".

Potrebbe inoltre essere possibile personalizzare la visita in base alla scelta di parametri quali la durata, il grado di approfondimento, l'argomento, il periodo storico, il tipo di percorso, ecc. L'esplorazione completa di un vasto sito potrebbe essere suddivisa in più percorsi illustrativi che l'utente potrebbe intraprendere in momenti diversi, analogamente a quanto accade con alcune delle più evolute guide audio che sono noleggiabili all'ingresso di alcuni musei.

Una volta conclusa una visita preregistrata, l'avatar automatico può congedarsi, l'utente si ritrova al punto di partenza e, se lo desidera, può intraprendere una nuova esplorazione.

### **6.1.3 Consulenza**

In contesti in cui l'ambiente collaborativo viene utilizzato per scopi di e-commerce, può essere utile un servizio di consulenza fornito da una guida.

La possibilità di avere una consulenza da parte di un altro utente è tanto più interessante quanto più la tipologia di articolo offerta attraverso il sistema di e-commerce è specialistica, o richiede un elevato grado di personalizzazione, e quindi la procedura di acquisto o di scelta del prodotto diventa particolarmente complessa.

In generale, pertanto, laddove non sia disponibile una persona reale che possa fungere da consulente all'acquisto on-line, è possibile sostituire un agente, che funga da



## **6.2 Implementazione di un sistema assistivo alla collaborazione**

Sulla scorta delle osservazioni precedenti, è stato implementato un sistema assistivo alla collaborazione integrabile nell'architettura generale del nostro sistema collaborativo tridimensionale.

Le funzionalità realizzate ricadono nelle tre principali categorie in cui è possibile suddividere un canonico sistema assistivo: funzionalità di aiuto; funzionalità di guida automatica; popolamento e dimostrazione.

### **6.2.1 Funzionalità di Aiuto**

Le funzionalità di aiuto servono a portare assistenza all'utente in difficoltà nell'utilizzo dell'applicazione, oppure a fornire informazioni riguardo ai contenuti. L'aiuto può essere portato in due modalità fondamentali: in modo propositivo ed in modo reattivo.

Nella prima modalità, l'agente rileva alcuni parametri che indicano chiaramente uno stato di difficoltà dell'utente, ed interviene automaticamente prestando aiuto. Tipici parametri sono un tempo di arresto eccessivo nel medesimo punto, la ripetizione di un comando errato un numero eccessivo di volte, la prossimità ad un oggetto interattivo senza che l'utente interagisca effettivamente con l'oggetto, ecc.

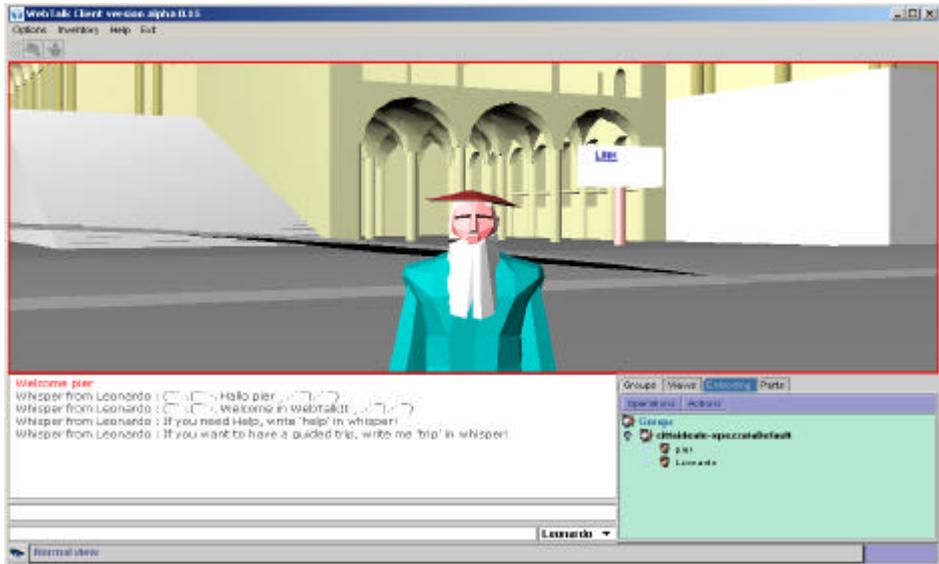
Nella seconda modalità, quella reattiva, l'agente arriva in aiuto dell'utente in risposta ad una esplicita chiamata di aiuto mediante un particolare comando ed azione.

È necessario un buon equilibrio tra queste due modalità di interazione per evitare all'utente la sensazione che l'aiuto sia troppo *invadente*, o che al contrario sia troppo difficile da raggiungere. In altre parole, deve essere propositivo al giusto grado, e velocemente reattivo.

Una prima funzionalità dell'agente propositivo è l'individuazione dell'ingresso di un nuovo utente, e l'esecuzione di un saluto. Lo scopo del saluto sta nel rendere immediatamente noto all'utente che si trova all'interno di una applicazione collaborativa che ospita anche altri utenti (eventualmente robotizzati).

L'azione di saluto prevede che l'agente determini qual è il frustum di visualizzazione attuale dell'utente (in altre parole, determini geometricamente qual è il suo campo visivo), vi si ponga all'interno, e mandi una comunicazione scritta di saluto e benvenuto, utilizzando eventualmente anche il nome con cui l'utente si è collegato al sistema.

In generale l'agente non può *percepire visivamente* gli altri avatar – come avviene invece tra utenti umani - e sfrutta necessariamente informazioni geometriche per tenere traccia delle posizioni di tutti gli oggetti che lo circondano. Ad ogni nuovo accesso, l'agente deve posizionarsi in maniera tale da poter rientrare nel campo visivo del nuovo utente ed essergli rivolto di fronte. In tal modo può presentarsi e porre il messaggio di benvenuto.



**Fig. 6-4: Immagine del sistema collaborativo tridimensionale in esecuzione: l'agente si pone nel campo visivo del nuovo utente automaticamente, e gli porge un messaggio di benvenuto. In questa particolare applicazione, l'agente è rappresentato da un'immagine di Leonardo da Vinci.**

Appena l'utente accede al mondo e si muove all'interno dello spazio condiviso, l'agente ricalcola la propria posizione e la aggiorna continuamente in modo da rimanere visibile almeno per i primi 10 spostamenti registrati dopo l'ingresso. In questo modo l'utente vede l'agente mentre ha la possibilità di leggere le frasi di benvenuto che questi gli invia in chat. Dopo questa fase iniziale, l'agente interrompe il suo movimento e lascia l'utente libero. Se continuasse a posizionarsi in prossimità potrebbe diventare fastidioso: il meccanismo deve funzionare solo fino a che l'agente abbia presentato i primi saluti.

L'agente spiega anche in che modalità richiamarlo se si desidera il suo aiuto in un secondo momento (mediante la digitazione del comando 'help' sulla linea di conversazione): in questo modo si invoca aiuto dall'agente attivandone il *comportamento reattivo*.

Un'ulteriore modalità di funzionamento propositivo dell'agente di aiuto consiste nell'ascolto sul canale di conversazione di frasi del tipo:

*"Hello", "Hello people", "anybody there?"*, che caratterizzano disorientamento da parte dell'utente o ricerca di altri utenti con cui collaborare. In risposta a queste forme di interazione da parte dell'utente, l'agente si ripropone, ricordando che è disponibile per fornire aiuto o per cominciare una visita guidata.

L'agente interviene in risposta ad altre brevi forme di comunicazione. Ad esempio quando un utente si allontana momentaneamente dal discorso che scorre in chat, di norma è d'uso utilizzare l'acronimo *brb* (*be right back*). In questo modo gli altri partecipanti percepiscono l'allontanamento di uno degli interlocutori dal contesto. È indice di cortesia inviare saluti all'uscita ed anche al rientro in chat. Questo semplice compito è affidato all'agente.

*Thimoty* -> '*brb*'

*Agent* -> '*see you soon, Thimoty!*'

...

*Thimoty* -> '*back*'

*Agent* -> '*Welcome back Thimoty!*'

*Agent* -> '*We were getting bored without you!*'

...

Da osservazioni empiriche è emerso che più del 50% del tempo di permanenza nel sistema viene normalmente impiegato nel gestire i movimenti dell'avatar<sup>24</sup> attraverso le geometrie dei mondi: l'esplorazione dell'ambiente sembra essere la prima attività che cattura l'attenzione dell'utente, di conseguenza l'osservazione dei movimenti degli avatar può aiutare a capire se siano insorte o meno difficoltà di utilizzo del sistema.

Se l'utente non è alla sua prima esperienza, tutto prosegue normalmente; se invece riscontra difficoltà potrebbe rimanere bloccato, e l'agente automatico, monitorando le variazioni di posizione, rivela la situazione di stallo e decide di intervenire riproponendo i suggerimenti e fornendo un insieme arricchito di istruzioni inerenti il problema.

In realtà l'utente potrebbe essere impegnato in altra attività, ad esempio in conversazione via chat, in consultazione di contenuti 2D nel Web o anche semplicemente potrebbe aver abbandonato la postazione di lavoro momentaneamente.

Alcune di queste situazioni possono essere rivelate: se l'utente è impegnato in chat, l'agente se ne accorge monitorando e analizzando i messaggi di chat che riceve, e può decidere di non intervenire. Se a tutto ciò si aggiunge una adeguata temporizzazione degli interventi si può evitare un comportamento invadente dell'agente: se alla proposta di aiuto non succede, in un breve lasso temporale, una qualche richiesta dell'utente, l'agente sospende l'invio di messaggi e resta comunque pronto ad intervenire nel caso di richieste esplicite.

In modalità *reattiva* l'agente riceve un certo numero di comandi dalla linea di conversazione, eventualmente strutturati gerarchicamente all'interno di un menu con struttura ad albero.

In particolare, il comando di *help* è ulteriormente definibile specificando, dopo la parola *help*, l'argomento su cui si desidera aiuto o informazioni. Ad esempio, digitando *help fly*, si ottengono informazioni su come esplorare il mondo volando. Le possibili combinazioni di richiesta di aiuto sono suggerite dall'agente al momento dell'invocazione.

---

<sup>24</sup> T. Barbieri e P. Paolini – Cooperative Visit for Museum Sites a Year Later: Evaluating the effect - *In proceedings Museum&Web 2000 – Minneapolis (USA)*

Le risposte alle richieste d'aiuto e eventuali informazioni su ulteriori comandi accettati sono inviate dall'agente nell'area di comunicazione testuale. Nella comunicazione testuale remota, spesso accade che i messaggi particolarmente lunghi vengano spezzati su più linee. Anche se ciò non è strettamente necessario, per ridurre il ritardo nella comunicazione dovuto al tempo di digitazione del testo, si digita la seconda linea di testo mentre la prima è stata già inviata. Nel caso dell'agente vengono spezzate le frasi lunghe su più linee, per facilitarne la lettura, e viene introdotto un meccanismo di ritardo temporale per dare l'impressione che le frasi siano pensate o scritte al momento. Una eccessiva prontezza nella risposta in un ambiente testuale, infatti, può essere disorientante<sup>25</sup>.

L'ambiente di chat è ideale per le comunicazioni di brevi messaggi, e i contenuti che è possibile trasmettere all'utente sono di entità molto ridotta rispetto a ciò che si potrebbe ottenere consultando un sistema di help in linea tradizionale. Ciò non di meno la comunicazione nella forma *domanda- risposta* può risultare sufficientemente esaustiva nel contesto delle brevi spiegazioni che l'utente può richiedere, conservando la freschezza di una interazione diretta.

## 6.2.2 Funzionalità di guida automatica

La funzionalità di guida automatica viene attivata mediante esplicita richiesta da parte dell'utente dalla linea di comando.

In particolare vengono forniti due comandi: 'trip' attiva un percorso guidato di navigazione privato, e 'public trip', in cui invece la descrizione del percorso guidato avviene su un canale di conversazione pubblico.

La distinzione sta nel fatto che nel caso di visibilità privata, l'agente dialoga con il suo interlocutore su un canale di comunicazione privato e non disturba l'area di chat condivisa da tutti gli altri utenti del sistema collegati in quel momento. Se la visita è invece collettiva (più di un utente) allora si rende necessario che il contenuto del testo sia inviato contemporaneamente a tutti gli utenti. In questo caso, l'utente che effettua l'esplicita richiesta diviene il punto di riferimento dell'agente per la ricezione dei comandi, cioè può decidere se interrompere la gita, se passare alla prossima tappa, ecc. In questo modo si evita che l'agente debba eseguire comandi diversi e discordanti provenienti da tutti i membri del gruppo.

Un'altra ipotesi di gestione della visita guidata intrapresa da più utenti contemporaneamente è quella dell'applicazione della metafora cooperativa dei gruppi. Un utente può creare un gruppo o decidere di aggiungersi ad uno già esistente. Una volta formato, un gruppo è come un sottoinsieme degli avatar che popolano il mondo in quel momento. Gli utenti del gruppo comunicano in una sorta di canale riservato, ossia nella loro area di chat condivisa non vengono più visualizzati i messaggi degli utenti non appartenenti al gruppo. Questo può favorire la conduzione automatica durante la

---

<sup>25</sup> Pavel Curtis, "Mudding: Social Phenomena in Text-Based Virtual Realities", *Proceedings of DIAC '92*, which is available via anonymous ftp from [parcftp.xerox.com/pub/MOO/papers/DIAC92](http://parcftp.xerox.com/pub/MOO/papers/DIAC92).

visita perché l'isolamento dal contesto globale permette che si crei un'area tematica. Se l'agente si unisse al gruppo o fosse egli stesso a crearne uno dedicato alla visita guidata, la situazione che si verrebbe a creare sarebbe simile a quella di un museo, infatti in un museo più gruppi contemporaneamente possono seguire la loro guida e accedere ai contenuti senza interferire con altri gruppi impegnati nella medesima attività. Se l'utente non interagisce con l'agente per troppo tempo, scatta un dispositivo di time out che conclude la visita. L'agente saluta l'utente, si congeda e torna alla sua posizione iniziale.

La visita guidata che la guida automatica offre agli utenti che la richiedono è strutturata su un percorso diviso a tappe. Le tappe sono costituite da posizioni predefinite all'interno del contenitore ambientale 3D che rappresenta lo spazio di collaborazione (ad esempio, davanti ad oggetti o geometrie di particolare interesse ai fini della visita). In corrispondenza ad una particolare posizione spaziale, viene associata una descrizione testuale a diversi livelli di dettaglio (descrizione iniziale, dettagli di primo livello, dettagli di secondo livello), e la possibilità di saltare eventualmente a un link bidimensionale, che viene proposto nella comunicazione. Per ogni tappa di un percorso, è eventualmente presente un'informazione che specifica se dalla tappa presente è possibile proseguire per una visita guidata diversa (avente la tappa corrente in comune). In questo modo è possibile specificare all'interno del sistema collaborativo diversi percorsi guidati, che si distinguono, ad esempio, per la tematica affrontata, e per il diverso taglio di presentazione dei contenuti offerti nell'applicazione collaborativa.

La scelta di un percorso, il passaggio da una tappa all'altra, e la richiesta di maggiori informazioni testuali nell'ambito di ciascuna tappa, vengono effettuate dall'utente mediante l'invio degli opportuni comandi, di volta in volta proposti dall'agente durante la visita guidata.

I percorsi e le tappe sono descritti mediante un file XML rispondente ad un opportuno DTD.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.0 (http://www.xmlspy.com) -->
<!ELEMENT guided_tour (stage*,server_IP?,server_PORT?,agent_name,
agent_world,world_part,avatar_URL)>
<!ELEMENT server_IP (#PCDATA)>
<!ELEMENT server_PORT (#PCDATA)>
<!ELEMENT agent_name (#PCDATA)>
<!ELEMENT agent_world (#PCDATA)>
<!ELEMENT world_part (#PCDATA)>
<!ELEMENT avatar_URL (#PCDATA)>
<!ELEMENT stage (position,text,more_text*,mobile_object*,stage_path,stage*)>
<!ELEMENT position (#PCDATA)>
<!ELEMENT text (#PCDATA)>
<!ELEMENT more_text (#PCDATA)>
<!ELEMENT mobile_object (#PCDATA)>
<!ELEMENT stage_path (#PCDATA)>
```

Il presente DTD lega un particolare agente (identificato da un nome e da un mondo di appartenenza) ad un particolare server (IP e porta specificate) per la distribuzione. Per

ciascun percorso (elemento ‘guided tour’) è possibile specificare una tappa, che contiene testo descrittivo, il percorso alla tappa successiva, e un eventuale riferimento ad un oggetto con cui è possibile interagire nell’ambito della tappa corrente.

Ecco un esempio di percorso guidato all’interno dell’applicazione di esempio “La città Ideale di Leonardo da Vinci”.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE guided_tour SYSTEM "Path.dtd">
<guided_tour>
  <server_IP>131.175.167.47</server_IP>
  <server_PORT>6764</server_PORT>
  <agent_name>Leonardo</agent_name>
  <agent_world>cittaidealespezzata</agent_world>
  <world_part>Central2</world_part>
  <avatar_URL>http://hoc115.elet.polimi.it/geometries/MAN_0.wrl </avatar_URL>
  <stage>
    <position>143#12#-77</position>
    <text>Look at this beautiful closter! Leonardo was a great architect!</text>
    <more_text>In the closter we count more then one hundred windows!!!</more_text>
    <stage_path>stage01.hag</stage_path>
  </stage>
  <position>196.22968#10.335736#-115.89712</position>
  <text> Another machine of Leonardo: "The Motion Trasformer"</text>
  <more_text>You can interact whit this object by usign your mouse!>
  </more_text><more_text>You can interact also turning the handler!</more_text>
  <mobile_object>Traf-fisso</mobile_object>
  <mobile_object>Traf-Ruota1</mobile_object>
  <stage_path>stage03.hag</stage_path>
</stage>
</stage>
<stage>
  <position>236.30472#33.357132#-67.35333</position>
  <text>This is the Crane, one of the machines of Leonardo</text>
  <more_text>You can move the two rudders...</more_text>
  <mobile_object>Gru-Mobile</mobile_object>
  <mobile_object>Gru-Fissa</mobile_object>
  <mobile_object>Gru-Ruote</mobile_object>
  <stage_path>stage02.hag</stage_path>
</stage>
</guided_tour>
```

La creazione dei file XML che descrivono i percorsi degli agenti guida avviene attraverso una opportuna applicazione, la quale funge da *registratore* di eventi. E’ possibile navigare all’interno dell’applicazione e registrare i movimenti effettuati all’interno del mondo all’interno di file di testo in particolare formato (.hag). Questi file di registrazione dei movimenti sono poi collegati a ciascuna tappa di percorso all’interno del file XML, per descrivere il movimento effettivo dell’agente per raggiungere la successiva posizione.

L’utilizzo di percorsi preregistrati semplifica di molto l’implementazione dell’agente guida, non dovendo costringere l’agente ad analizzare la topologia della geometria ambientale in cui si trova (e dei relativi ostacoli disseminati in essa) per trovare autonomamente un percorso di collegamento adatto. Tuttavia l’utilizzo di percorsi

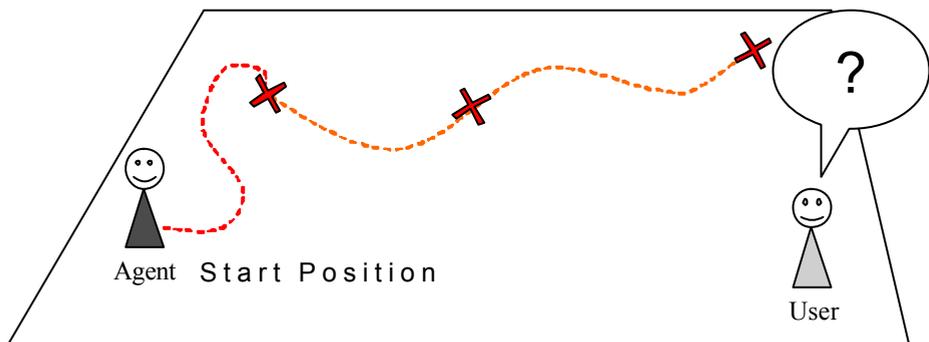
preregistrati pone alcuni problemi dal punto di vista dell'interazione con l'utente. L'agente, infatti, è attivo anche per richieste di aiuto e, come già illustrato, nel caso che riceva un messaggio che lo riguarda direttamente, ricalcola le proprie coordinate e si posiziona di fronte all'utente che lo ha invocato.

Questo accade per qualsiasi posizione assuma in quel momento l'utente. Oltre a ciò vi sono altri meccanismi che attivano l'agente, come la prossimità degli avatar ad oggetti animati. Se ad esempio un avatar passa nelle vicinanze di una geometria mobile posta all'interno del mondo, l'agente si avvicina e mostra che è possibile sperimentare le animazioni previste per quel oggetto, in modo da invogliare l'utente a provare di persona. Questo può portare l'agente a dislocarsi ovunque nel mondo.

Quando viene richiesto l'inizio di un tour guidato, l'agente è costretto a seguire la programmazione di movimento pre-registrata, e pertanto esiste un momento in cui è costretto a saltare dalla posizione corrente in cui si trova, alla posizione iniziale del percorso iniziale che è pre-registrato, creando una forte discontinuità spaziale che invariabilmente confonde l'utente.

La soluzione più semplice a questo problema potrebbe essere quella di spostare al punto di partenza del percorso guidato non solo la guida automatica stessa, ma anche tutti gli utenti che hanno fatto richiesta di una visita guidata. Tuttavia la possibilità tecnica di realizzare questa funzione dipende dall'architettura implementativa dello specifico agente.

Qualora l'agente di guida sia implementato come un emettitore di eventi allo stesso livello di un qualsiasi altro client, non risulta possibile forzare altri client a posizioni diverse da quelle correnti senza l'intercessione del gestore dello stato condiviso.



**Fig. 6-5: Un problema di usabilità nei tour guidati preregistrati: all'inizio del tour, l'agente è costretto a posizionarsi al punto di partenza iniziale creando una discontinuità spaziale. L'utente rimane disorientato se all'avvio del percorso guidato l'agente si allontana improvvisamente per portarsi alla posizione iniziale.**

Un altro problema derivante dallo sviluppo dei percorsi guidati è l'interazione con l'utente lungo i percorsi. Una volta avviato, l'agente procede verso la prima tappa. L'utente deve seguire l'avatar dell'agente attraverso le geometrie 3D del mondo. Questo comporta che in un ambiente ben realizzato e quindi sufficientemente complesso, ci siano scale da salire, angoli da svoltare e oggetti che possono parzialmente occultare la visione.

La navigazione 3D il cui scopo è l'inseguimento di un altro utente, dunque, può risultare piuttosto ardua, se non si prevedono dei meccanismi di correzione nel controllo dell'agente guida.

Vengono utilizzati due meccanismi di correzione:

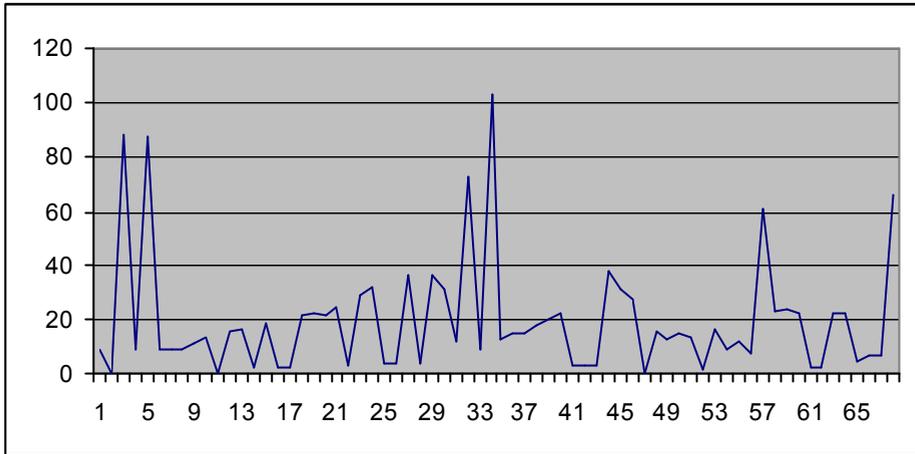
- controllo del frustum dell'utente: dalla posizione e dall'orientamento dell'avatar dell'utente, si determina il suo campo di visualizzazione. In questo modo è possibile calcolare se il successivo movimento dell'agente guida lo pone al di fuori del frustum di visualizzazione dell'utente a cui è collegato
- controllo della distanza: si calcola la distanza tra la posizione occupata dall'avatar guida e quella occupata dall'utente. Quando questa distanza supera un valore soglia, si arresta il movimento dell'avatar, consentendo all'utente di "recuperare".

L'utilizzo di un sistema di controllo della distanza rende più efficace la navigazione guidata.

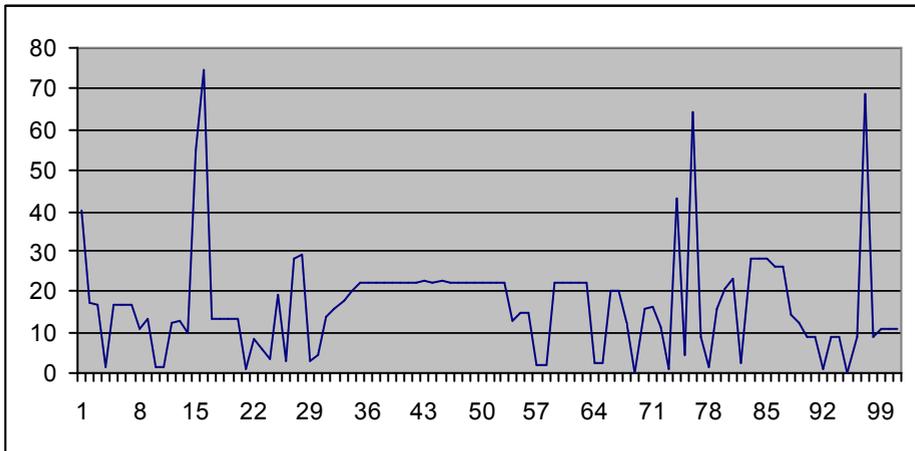
Raffrontando due navigazioni compiute seguendo la guida automatica da parte di un campione di utenti, una navigazione con agente privo dei meccanismi di correzione, e la seconda navigazione utilizzando i meccanismi di correzione, è possibile notare che la distanza tra gli utenti risulta più costante e regolare. Al contrario, senza meccanismi di correzione si notano dei picchi che corrispondono ai momenti in cui l'utente ha perso di vista l'agente guida.

### **6.2.3 Funzionalità di popolamento**

La terza funzionalità presente in un sistema assistivo è la funzionalità di popolamento e dimostrazione. Lo scopo è di creare l'impressione che l'ambiente virtuale sia popolato, ed utilizzare le "comparse" robotizzate (dette 'puppet') per dimostrare alcune funzionalità di navigazione all'interno del mondo, come ad esempio la possibilità di navigare volando, la possibilità di entrare o uscire da alcuni passaggi, di raggiungere certe parti della struttura tridimensionale, etc.



**Fig. 6-6: Inseguimento di un agente guida privo di meccanismi di controllo. In ascissa è riportato il tempo di navigazione, ed in ordinata la mutua distanza agente-utente.**



**Fig. 6-7: Inseguimento dell'agente guida dotato dei meccanismi di controllo. Il plateau nelle distanze ed il minor numero di picchi indicano un miglioramento nell'inseguimento.**

I puppet devono implementare un movimento apparentemente casuale e dare l'impressione di essere la rappresentazione grafica della presenza di altri utenti intenti nell'esplorare i mondi.

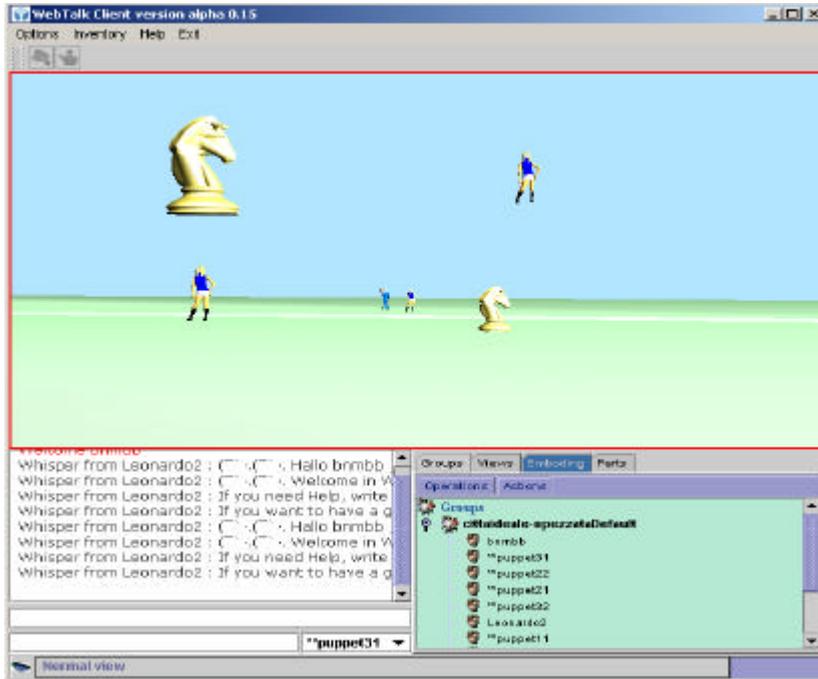


Fig. 6-8: Agenti Puppet in azione.

La loro rappresentazione deve avvenire con avatar di diversa forma e nel loro moto deve essere introdotto un errore casuale il cui scopo è far sembrare che la figura sia *animata*, infatti ad uno osservatore deve apparire che questa *fluttui* oscillando nell'aria perché il loro scopo è di rendere animato l'ambiente. La loro presenza può anche essere contestuale all'ambiente nel quale vengono inseriti: se ad esempio essi devono popolare la Città Ideale di Leonardo, è bene che vengano rappresentati con figure di personaggi abbigliati con abiti d'epoca.

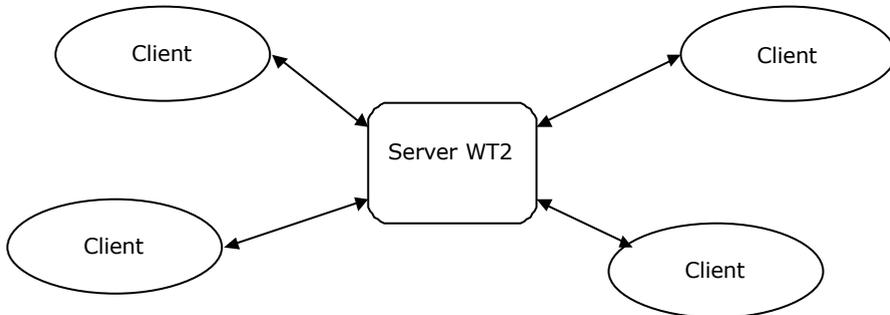
I puppet sono fatti girare come client indipendenti, che leggono dei movimenti preregistrati attraverso la medesima procedura adottata per gli agenti guida. Al tracciato di navigazione viene aggiunto un errore random alla posizione ed un ritardo variabile tra due successive variazioni di posizione, al fine di conferire un aspetto più naturale alla navigazione, e cercare di rendere meno evidente agli utenti la natura programmatica dei puppet.

## 6.2.4 Architettura generale dei moduli agente

Una generale architettura per sistemi collaborativi può adottare varie strategie. Tuttavia la strategia di riferimento scelta per la maggior parte delle funzionalità (nell'architettura denominata WebTalk-II) riguarda la realizzazione di una comunicazione client-server, con il client preposto alla presentazione dell'istanza locale dello spazio condiviso, ed il server preposto alla gestione e sincronizzazione dello stato condiviso.

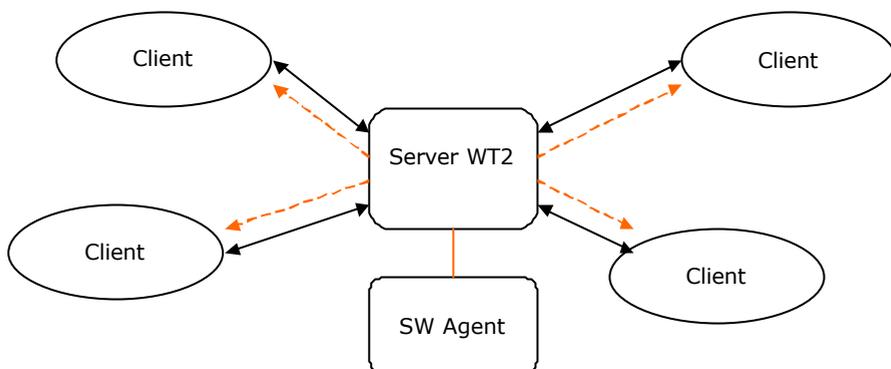
In quest'ottica, esistono fondamentalmente due scelte architetture per la realizzazione dei moduli che controllano gli agenti per la collaborazione assistita: la prima scelta riguarda l'implementazione dell'agente come se fosse un modulo aggiuntivo che opera lato server verso i client; la seconda scelta prevede invece una implementazione a livello client, senza che il server abbia possibilità di riconoscere se il client che riceve corrisponde ad un utente reale o uno automatico.

Nella soluzione architetture che vede gli agenti implementati lato server, il server ha a che fare con tutte le informazioni riguardanti i client di cui gestisce le connessioni in un dato momento. Le posizioni di qualunque avatar devono passare per il server per poter essere notificate a tutte le postazioni remote.



**Fig. 6-9: rappresentazione dell'interazione tra Client e Server nell'architettura di WebTalk-II.**

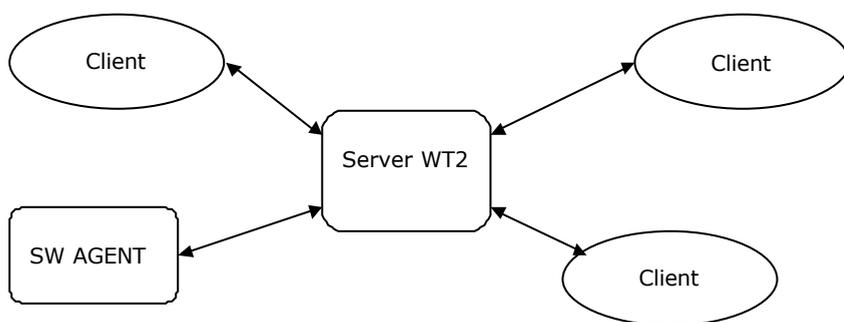
I moduli degli agenti necessitano di informazioni come la posizione degli avatar degli utenti per poter calcolare il loro posizionamento in automatico. Una soluzione architetture potrebbe essere quella di affiancare i moduli software degli agenti a quelli del server.



**Fig. 6-10: Posizionamento dei moduli agente a lato Server nell'architettura WebTalk-II. I pacchetti generati dagli agenti vengono inviati verso tutti i Client.**

Questi generano le informazioni inerenti gli agenti e le passano direttamente al Server che si occupa di notificarle ai client come se provenissero da un altro client, mentre in realtà il calcolo avviene all'interno del server.

In una visione alternativa, il software degli agenti automatici potrebbe trovarsi su una postazione remota e connessa al Server come qualunque altro Client. In questo modo al lato Server è completamente trasparente il fatto che i pacchetti che vengono ripetuti siano generati da un software o da un operatore umano che pilota il proprio avatar.



**Figura 6-11: Posizionamento dei moduli agente a lato Client nell'architettura WebTalk-II. I pacchetti generati dagli agenti vengono inviati verso il Server e quindi ripetuti come quelli di ogni altro Client WebTalk-II.**

I moduli agente possono essere lanciati su una macchina remota, essi ricevono dal Server i dati riguardanti gli altri avatar connessi (posizione, testo in chat, ecc.), li analizzano, e producono il comportamento dell'agente. Gli altri Client visualizzano l'agente come fosse un qualsiasi altro utente.

Le due ipotesi architetturali poste a confronto presentano alcune differenze. Il posizionamento della logica che controlla gli agenti a lato server è sicuramente più potente e versatile, e inoltre permette di ottimizzare il sistema. La maggiore potenzialità si può spiegare con il fatto che esistono eventi che un Client non può sollevare ed alcuni che può sollevare solo per se stesso, come ad esempio il passaggio da una parte del mondo ad un'altra, ed il conseguente innesco delle procedure di smontaggio e rimontaggio delle geometrie ambientali lato client.

Un altro problema difficilmente gestibile con una architettura sviluppata a lato Client è quello della replicabilità degli agenti. Se questi vengono connessi come Client al Server, per ogni agente è necessario gestire una connessione e ripetere i pacchetti in rete. Nel caso dei puppet non è di alcuna utilità inviare loro la notifica degli eventi prodotti dal mondo circostante e dagli altri avatar. I puppet non sono dotati di logica preposta all'analisi dei dati provenienti dagli utenti ed ignorano le informazioni di notifica del server, che però appesantiscono lo stesso il carico di rete.

Se fossero implementati a lato Server, il carico di rete verrebbe alleggerito perché la generazione dei pacchetti avviene in locale e la spedizione può essere eseguita solo verso i Client che necessitano degli aggiornamenti. Per fare questo è però necessario aggiungere sul Server la possibilità di gestire oggetti animati condivisi di tipo diverso, cioè avatar per i quali non occorre una operazione di connessione e caricamento.

Un ulteriore aspetto riguardante l'ottimizzazione che è possibile ottenere con una implementazione a lato Server è la possibilità di inviare in maniera selettiva i pacchetti. Nel caso che l'agente abbia iniziato una visita guidata con un utente, sarebbe possibile fare in modo che i pacchetti che permettono di visualizzare l'agente in movimento vengano inviati solo al Client o al gruppo di utenti attualmente impegnato nel percorso guidato. Questo creerebbe una inconsistenza nelle rappresentazioni dei mondi sui diversi Client collegati, ma risparmierebbe tutto il carico di rete necessario a inviare i dati del percorso dell'agente a Client che non sono coinvolti. Per attivare questa politica, tuttavia, occorre che il lato server abbia possibilità di riconoscere la natura automatica del client, rinunciando al concetto di completo disaccoppiamento tra server ed agente.

In definitiva la soluzione architetturale lato Server sembra essere la migliore, tuttavia richiede che venga modificata la struttura del Server per gestire la comunicazione interna con i moduli software degli Agenti.

Lo sviluppo degli agenti a lato Client, invece, può avvenire liberamente ed essere integrato come modulo aggiuntivo, senza necessitare di alcuna modifica dei moduli server esistenti. Alla luce di queste condizioni la scelta fatta per l'implementazione del prototipo per l'analisi delle caratteristiche della collaborazione assistita è stata quella

dell'architettura lato Client. Sulla base dell'esperienza acquisita potrà essere possibile in futuro l'integrazione dei moduli agente con i moduli Server.

### **6.3 Conclusioni**

Il concetto di collaborazione assistita è una novità introdotta dall'osservazione del comportamento degli utenti all'interno dei sistemi tridimensionali collaborativi. Partendo dalla semplice considerazione che la collaborazione tra utenti avviene solo all'interno di un gruppo, e che, quando essa si innesca, si autoalimenta coinvolgendo altri utenti, le tecniche di collaborazione assistita mirano a fornire un livello minimo di collaborazione che inneschi questo ciclo di autoalimentazione.

Dagli esperimenti eseguiti in laboratorio con i prototipi di collaborazione assistita realizzati, risulta che la loro funzione, oltre ad essere concettualmente indispensabile, è essenziale anche dal punto di vista pratico. Le modalità offerte di aiuto e assistenza dell'utente sono immediatamente comprese ed utilizzate, e le tecniche di dimostrazione diretta delle funzionalità del sistema sono tra i metodi più semplici di spiegazione dei meccanismi di collaborazione.

L'implementazione effettiva dei meccanismi assistivi ha dimostrato come la loro realizzazione efficace sia tutt'altro che banale e presenti numerosi problemi, sia dal punto dell'usabilità che dal punto di vista della realizzazione tecnica. Esempi di questa osservazione stanno nella difficoltà nel creare guide che possano conversare in modo non banale con l'utente, o che possano trovare un percorso di navigazione corretto all'interno di una geometria tridimensionale complessa, evitando di ricorrere alla registrazione dei percorsi.

Risulta tuttavia chiaro come ulteriori ricerche nel ramo della collaborazione assistita consentiranno di potenziare il livello di collaborazione raggiungibile all'interno degli spazi tridimensionali condivisi, e di consentire lo stabilirsi di collaborazioni efficaci tra utenti in tempi sensibilmente minori.

## 7 La multicanalità e l'uso di device mobili

### 7.1 *Fondamentali caratteristiche della multicanalità in contesto mobile*

#### 7.1.1 **La pila della multicanalità**

L'architettura collaborativa da noi esposta nei capitoli precedenti si concentra essenzialmente sull'utilizzo della rappresentazione tridimensionale per valutare come essa possa supportare in modo più efficace la collaborazione.

Purtuttavia, l'utilizzo del tridimensionale pone una barriera di ingresso significativa all'applicazione, sotto vari aspetti.

Il primo aspetto, come si è trattato nel capitolo precedente, è che non risulta per nulla semplice, in fase iniziale, comprendere come utilizzare la navigazione tridimensionale, e associare ad alcune metafore collaborative tipiche della navigazione tridimensionale l'effettiva modalità collaborativa che si utilizza in contesti reali. A questo scopo abbiamo proposto, come soluzione, l'utilizzo degli agenti virtuali.

Il secondo aspetto riguarda invece l'accessibilità effettiva ad un sistema che richiede elevata potenza di calcolo per il tracciamento della rappresentazione tridimensionale, ed una veloce connessione di rete per il mantenimento dello stato condiviso. Una soluzione parziale a questo problema può essere data dall'uso di tecnologie semplificate, ad esempio l'utilizzo di un plug-in VRML all'interno di un browser, come implementato nell'architettura da noi denominata WebTalk-I.

Più in generale, l'utilizzo dell'architettura del sistema collaborativo fin qui delineata presuppone l'accesso ad un calcolatore, escludendo quindi dalla collaborazione tutte le persone che si trovino in contesti precari, privi di un calcolatore, o addirittura in movimento o in viaggio.

L'idea che un'applicazione sia fruibile non solo attraverso una singola tecnologia di accesso, sta alla base di un concetto che sta acquistando sempre maggiore significato grazie allo svilupparsi di diverse tecnologie e infrastrutture di comunicazione: la multicanalità.

La nozione generale di *canale* comprende tutte le architetture di comunicazione, i protocolli, i dispositivi fisici, l'hardware ed il software necessari per utilizzare una applicazione il cui scopo sia il trattamento dell'informazione, la sua navigazione, o la sua modifica. La *multicanalità* prevede dunque una fornitura di accesso ad una stessa applicazione informativa mediante un insieme di canali diversi, ma dando la possibilità di fruire della medesima informazione, almeno in principio.

Esempi di applicazioni multicanale in rapida diffusione sono i servizi finanziari o di trading offerti da numerosi istituti di credito. Mediante canali diversi quali l'uso del

chiosco elettronico direttamente nella sede della banca, l'accesso al web, l'accesso mediante il telefonino WAP, oppure ancora l'accesso web mediante un set-top-box collegato al televisore di casa, è possibile accedere alle informazioni del proprio conto corrente e del proprio portafoglio titoli, ed effettuare operazioni, sui medesimi dati, quale che sia il canale di accesso che si decide di utilizzare.

Il rapido maturare da un lato della comunicazione radiomobile (GSM, GPRS e UMTS), sia dal punto di vista delle prestazioni tecnologiche che della capillarità della loro presenza sul territorio (almeno nella realtà italiana attuale), e dall'altro delle periferiche mobili - cellulari e palmari - sempre più potenti in termini di prestazioni e dall'autonomia sempre maggiore, rendono l'uso del canale mobile una realtà di sempre maggiore interesse.

Mediante l'uso di un cellulare evoluto o di un palmare connesso ad una tecnologia radiomobile a larga banda, le possibilità di interazione con applicazioni informative a tradizionale accesso da postazione fissa diventano sempre più concrete.

Dal punto di vista dell'utente si sta verificando dunque una convergenza tra la Internet fissa e quella mobile, cui è possibile accedere attraverso una device mobile ed una rete radiomobile. L'integrazione tra i due principali oggetti protagonisti di questa rivoluzione della comunicazione, il telefono cellulare da una parte, ed il computer palmare dall'altra, sta avvenendo a grandi passi.

Saranno questo tipo di periferiche, i cellulari in grado di collegarsi ad Internet mediante il protocollo WAP, ed i palmari connessi ad un modem cellulare, a costituire il campo di indagine nel proseguimento del capitolo, per tentare di analizzare come lo spazio condiviso di collaborazione possa essere allargato al contesto mobile verso utenti dotati di canali di questo tipo, quali siano le differenze da tenere in conto, e come possa essere valutato il risultato finale.

Si è detto come nel concetto di *canale* ricadano sostanzialmente le tecnologie di connessione utilizzate per il collegamento ad una medesima applicazione, al fine di consentire accessi di natura eterogenea alle stesse informazioni. Se tuttavia le informazioni e il nucleo dell'applicazione fruita rimangono in questa prospettiva del tutto invariate, forzatamente diverse dovranno essere le modalità con cui le informazioni stesse sono inviate in ciascun canale, e l'aspetto stesso dell'applicazione assumerà in generale forme diverse.

Il canale, dunque, è un concetto che porta con sé non solo una cifra tecnica, ma pone delle scelte su quattro importanti aspetti:

- la *forma* della comunicazione, intendendo con questo il modo con cui l'informazione è presentata all'utente, e come le risposte dell'utente sono raccolte e riportate all'applicazione. Questo livello viene spesso anche definito "presentazione".
- la *navigazione* nell'applicazione, come cioè l'utente percepisce la struttura dell'applicazione e quali meccanismi gli sono forniti per interagire con essa ed

attivarne le funzionalità. Le modalità di navigazione sono legate alla presentazione che viene scelta dall'applicazione, e alle caratteristiche fisiche del canale.

- il *dominio funzionale*, legato a quali funzioni si riescono a presentare, e quali navigazioni ad implementare all'interno del canale. Laddove una particolare funzione necessita di un particolare tipo di interazione o di presentazione, non disponibile in un dato canale, la funzione corrispondente non può essere erogata sul quel canale. Pertanto il concetto di canale implica anche la scelta, in termini di aggiunta o di cancellazione, delle funzioni effettivamente disponibili all'interno dell'applicazione così come è percepita dall'utente
- l'aspetto *collaborativo*, vale a dire come l'utilizzo di un certo canale influisce sulle metafore di collaborazione all'interno dell'applicazione, sia tra utenti che utilizzano il medesimo canale, sia tra utenti che popolano l'applicazione provenendo da canali diversi.

Questi livelli di analisi formano una pila funzionale, in cui ciascun aspetto dipende dal precedente per la propria definizione. L'analisi in serie di ciascun livello della pila consente di comprendere l'effettiva natura del canale e di arrivare a definire un idoneo sistema di collaborazione.



**Fig. 7-1.** La 'pila della multicanalità' esprime le relazioni di dipendenza tra le principali caratteristiche di canale

### 7.1.2 Presentazione

Nell'affrontare il primo punto, la *presentazione*, l'approccio più corretto è quello di separare nettamente l'informazione contenuta effettivamente dall'applicazione dal modo con cui essa deve essere presentata, essendo la presentazione dipendente dal canale. Proprio per questo scopo sono stati definiti molti dei nuovi standard W3C: XML, XSL, XSLT.

La possibilità di rappresentare in XML la struttura semantica dell'informazione, e specificare per essa più trasformazioni (XSL, XSLT), una per ciascun canale, consente di generare in modo automatico più presentazioni, in formati diversi (HTML, PDF, WML, VXML, ecc.), per canali diversi. Così il risultato di una interrogazione su una base dati rappresentato in XML, può generare, in rapporto alla richiesta, una pagina HTML per un browser web, un documento PDF per una stampante di rete, una card WML per un telefono cellulare, una pagina VXML per un sintetizzatore vocale che parla ad un utente che sta conversando a voce con il servizio.

Al giorno d'oggi esistono numerosi browser per dispositivi handheld: Microsoft ha convertito il noto Internet Explorer per i computer che posseggono il sistema operativo Windows CE e sono stati implementati client per i PalmPilot. Questi browser sono molto simili ai loro corrispettivi formati desktop; essi possono gestire molti dei formati di informazione statica che risiede sulle pagine Web – immagini, tabelle, form. La sostanziale differenza rimane principalmente nelle dimensioni dei display di visualizzazione, aspetto che impatta grandemente sulla presentazione da adottare.

I maggiori problemi a livello di presentazione sono legati alla scarsa velocità di trasferimento dei dati, alle limitate capacità tecnologiche degli attuali display (per esempio la visualizzazione in scala di grigi e non a colori), alle dimensioni dei display. Questi due aspetti impediscono sia di mandare informazioni grafiche complesse, che richiederebbero un eccessivo tempo di trasmissione sui canali radiomobili di epoca pre-G2.5 e pre-G3, sia le scarse possibilità di resa visiva presso l'utente finale.

L'altro problema di presentazione riguarda la ridotta dimensione dei display, che non consente di rappresentare grandi quantità di informazione all'interno della stessa schermata. Se questo rappresenta un problema per visualizzazioni complesse, dotate di grafica e testo, su schermate di semplice testo, in una periferica munita di un efficiente sistema di scorrimento, il numero di righe contemporaneamente visualizzate a video non sembra essere un parametro che influenzi la rapidità di comprensione nella lettura del testo stesso.<sup>26</sup>

### 7.1.3 Navigazione

Gli effettivi meccanismi di *navigazione* discendono direttamente dalla forma di presentazione prescelta, e pertanto ciascun canale permette di esplorare le informazioni in modo diverso (in una presentazione HTML si seguono i link con dei click del mouse, in una card WML si è limitati ad usare il cursore su/giu del telefonino). Tuttavia in una realizzazione di una applicazione cui si prevede un accesso multicanale occorrerebbe che la navigazione fosse progettata avendo in mente alcuni obbiettivi:

- *Interscambiabilità delle sessioni*: consentire cioè ad una navigazione iniziata con un canale di proseguire mediante un altro canale qualora esso si trovi

---

<sup>26</sup> Dillon A., Richardson J., McKnight C. – *The effect of display size and text splitting on reading lengthy text from the screen - Behaviour and Information Technology*, 9(3) (1990)

disponibile, agganciando alla navigazione un concetto di stato che sia indipendente dal canale utilizzato

- *Valorizzazione delle caratteristiche del canale:* il sistema di navigazione dovrebbe essere progettato attorno agli aspetti di utilizzo più pratici del canale utilizzato, senza cercare di forzare aspetti navigazionali impossibili su canali che non sono in grado di gestirli
- *Utilizzo di logiche di navigazione multiple:* ciascun canale dovrebbe offrire diversi livelli di navigazione, ad esempio per utenti più esperti o meno esperti, dove per gli utenti più esperti le informazioni di aiuto, od i passaggi necessari ad effettuare la navigazione potrebbero essere concettualmente più complessi ma complessivamente più rapidi e meno assistiti. Un esempio è la possibilità di evitare di utilizzare il mouse per la navigazione in alcuni applicazioni, ricordandosi tutte le scorciatoie da tastiera, oppure evitare di visualizzare alcune pagine di informazione già note quando si trasmette ad un cellulare WAP.

Altri aspetti della navigazione sono strettamente legati alla fisicità del mezzo con cui si fruisce il canale. Continuando a descrivere problematiche riguardo a canali che utilizzano cellulari o palmari in reti radiomobili, le minori dimensioni del display costringono ad una più frequente interazione con il sistema di navigazione dello schermo, pertanto è opportuno che questo sia di facile accesso e di rapido utilizzo. Un esempio lampante è la presenza su tutti i cellulari moderni di un piccolo joystick o tasto cursore direzionale posto al centro, immediatamente sotto al display: tasto che si trova, in foggia leggermente diversa, anche sotto il display della maggior parte dei palmari. L'utilizzo della metafora dei menu a tendina, invece, non presenta problemi nel trasferirsi su display ristretti, posto naturalmente il vincolo di una massima dimensione del menu

Un altro aspetto è legato alla minore facilità di utilizzo di sistemi di input, dovuto ad esempio alla mancanza di una tastiera. Su un cellulare, l'inserimento di lunghe stringhe di testo è scomodo, a causa della necessità di cercare le giuste combinazioni sui tasti numerici, mentre sul palmare, occorre acquisire una certa dimestichezza con il software di riconoscimento dello stilo. In questa ottica, sono state sviluppate interfacce quale la T9, che tentano di ridurre il numero di micro-operazioni legate all'inserimento di un testo.

Specie in ambito di terminali cellulari, occorre fornire accesso quanto più possibile diretto alle risorse, senza costringere alla digitazione. L'accesso può avvenire fornendo un sistema di bookmark, oppure un menu di navigazione avanzato.

Paradossalmente, questo problema di navigazione si presenta identico nella realizzazione di un sistema di navigazione vocale utilizzando VoiceXML. Nei menu vocali, occorre sempre restringere le scelte di navigazione utilizzando una opportuna grammatica. Ogni volta che bisogna specificare del testo libero, spesso bisogna ricorrere ad un difficoltoso spelling lettera per lettera, con una elevata percentuale di errore nel riconoscimento.

Per ultimo, occorre cercare di ridurre al minimo i movimenti necessari per lo scrolling, anche se questo avviene mediante una funzione di facile utilizzo. Gli utenti sono potenzialmente portati ad effettuare ripetuti scorrimenti di una pagina visualizzata su un piccolo display; questo però distoglie l'attenzione dal principale obiettivo che essi si erano prefissi.

Lo scrolling può essere ridotto operando opportune scelte in fase di progettazione:

- Ponendo in un punto preciso della pagina un menu che sia di riferimento;
- Inserendo informazioni chiave all'inizio di una pagina;
- Riducendo la quantità di informazione per pagina, concentrandone il contenuto informativo in modo che sia meno verboso possibile.

#### **7.1.4 Dominio Funzionale**

Un tipico aspetto di variazione nel *dominio funzionale* deriva dalle limitazioni tecniche e dalle caratteristiche fisiche di ciascun canale. Da questo punto di vista, le soluzioni non sono molte, se non cercare di fornire eventualmente un servizio sostitutivo alla funzione mancante. Per fare un esempio, una funzione offerta all'interno di un canale Web HTTP-based è quella di salvare sul proprio disco locale le informazioni cercate e rinvenute sul web. La funzione di ricerca, per sé, è erogabile praticamente senza alcuna differenza anche attraverso il canale WAP, senza che in media la prestazione della ricerca, in termini di risultati ottenuti, soffra del cambiamento di canale<sup>27</sup>. Tuttavia, non disponendo un cellulare di un disco fisso e nemmeno di una zona di memoria per la memorizzazione temporanea, non risulta possibile con il canale WAP salvare un file localmente. Se quindi in questo caso l'aspetto tecnico sembrerebbe imporre una diminuzione del dominio funzionale, è vero invece che questa funzione può essere recuperata semplicemente fornendo un servizio di natura diversa, ad esempio la memorizzazione del file trovato su un server di rete raggiungibile via WAP, e l'attivazione di un bookmark sul proprio cellulare. In questo modo, seguendo il bookmark si legge automaticamente la pagina scaricata e pronta per l'utente dal server remoto.

Altre variazioni di dominio funzionale percepibili dall'utente sono strettamente legate alle possibilità offerte dall'interfaccia grafica in termini di presentazione e navigazione. Nel caso di una applicazione i cui dati sono costituiti da geometrie tridimensionali, ad esempio, è chiaro come nel contesto mobile, su periferiche povere in termini di rappresentazione grafica, ogni funzione legata alla manipolazione visuale di queste geometrie sia destinata ad essere in qualche modo surrogata od eliminata del tutto.

#### **7.1.5 Collaborazione Multicanale**

Anche le metafore di collaborazione sono legate, come detto, al particolare canale. In particolare, essendo la loro realizzazione sostanzialmente una commistione tra la strategia di presentazione, quella di navigazione, e il dominio funzionale offerto, risulta

---

<sup>27</sup> Norliza Mohd-Nasir, Kevin Boone, George Buchanan - Improving Web Interaction on Small Displays - in Proceedings of the 8<sup>th</sup> International World Wide Web Conference, Toronto, Canada, May 11-14 1999

chiaro come occorra arrivare a considerare quali siano le metafore collaborative offribili, solo dopo l'analisi dei primi tre aspetti relativi al canale in considerazione.

Una ulteriore complessità nel definire le possibili collaborazioni nasce dall'osservazione che anche la particolare applicazione offerta influisce sulla natura della collaborazione, ed anzi, essa è la linea guida da seguire per la progettazione delle effettive metafore di collaborazione. Così, le dinamiche collaborative di una piattaforma di elearning, possono essere sostanzialmente diverse da quelle necessarie per una sessione di visita museale, e via dicendo.

Nel particolare contesto mobile, il progetto della collaborazione tra utenti deve tenere conto delle particolari modalità con cui il canale viene utilizzato. Se ne possono identificare principalmente tre<sup>28</sup>:

- *Visiting*: consiste nel lavorare in differenti posti per un significativo periodo di tempo: in questo ambito è come se l'utente collaborasse in modo fisso per tratti di tempo brevi, e da un numero di stazioni diverse.
- *Travelling*: consiste nel lavorare in un veicolo, ad esempio un aereo o un treno, con tutte le possibili interruzioni momentanee che l'attività del viaggiare può causare, ma con la possibilità di interagire in modo continuativo per un periodo significativo;
- *Wandering*: consiste nel lavorare mentre ci si sposta fisicamente, e quindi avendo a disposizione un limitato livello di attenzione, una limitata possibilità di operare con le funzioni del sistema, ed un'elevata probabilità di frequenti disconnessioni dall'ambito collaborativo.

Da queste situazioni risulta chiaro come tra i meccanismi collaborativi in contesto mobile occorra senza dubbio prevedere modi per comunicare agli altri utenti il grado di partecipazione di ciascuno: se cioè, ad esempio, un particolare utente è connesso alla sessione condivisa ma non è in grado di parteciparvi attivamente per il momento.

La piattaforma di collaborazione generale presentata nei capitoli iniziali, abbraccia dunque quattro principali canali, tutti utilizzabili contemporaneamente per la collaborazione: un canale 2D che prevede la semplice navigazione web, un canale 3D all'interno di uno spazio virtuale condiviso, un canale mobile mediante l'uso di un cellulare wap, ed un ulteriore canale mobile, che fa invece utilizzo di computer palmari.

Lo studio dell'interazione di questi due ultimi canali con il canale tridimensionale, e la loro implementazione all'interno dell'architettura descritta è l'oggetto dei paragrafi successivi.

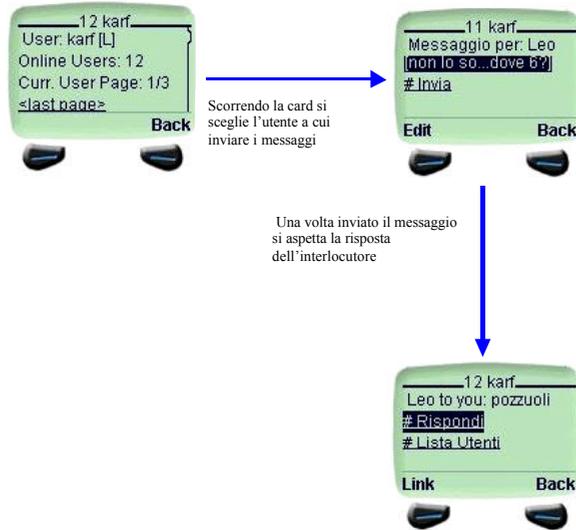
---

<sup>28</sup> S. Kristoffersen, F. Ljungberg – *Your Mobile Computer is a stationary computer* - in *Online Proceedings of the ACM Workshop on Handheld CSCW, Seattle, Washington, November 1998*

## 7.2 Implementazione del canale WAP

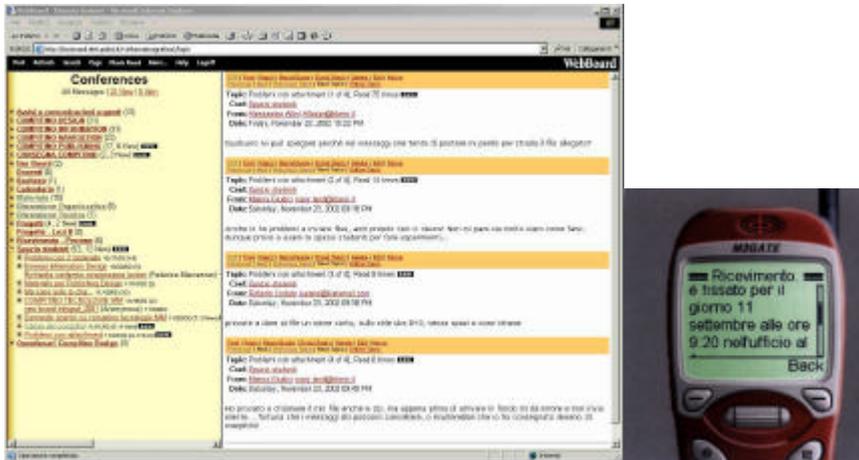
### 7.2.1 Esempi di collaborazione sul WAP

Sul canale WAP esistono vari tentativi di realizzare applicazioni che si spingono oltre la navigazione di card WML appositamente preparate, e che offrono un supporto collaborativo tra utenti. Un esempio tra i più diffusi sono i servizi chat, che creano un unico gruppo di utenti tra cui è possibile una conversazione senza particolari restrizioni.



**Figura 7-2. Una chat basata su WAP: gli utenti si scambiano messaggi navigando manualmente da una card all'altra, utilizzando gli opportuni pulsanti di navigazione forniti sul terminale cellulare.**

Altre applicazioni forniscono un front-end per siti bidimensionali di forum, consentendo un accesso mobile ai post del forum, ed eventualmente di eseguire dei nuovi post da cellulare. In questo modo risulta possibile una collaborazione il cui elemento cardine è la lettura e la scrittura dei messaggi del forum. Un esempio da noi realizzato riguarda l'interfacciamento WAP ad un famoso software per forum, WebBoard.



**Fig. 7-3: Al forum WebBoard viene agganciato un accesso attraverso un canale WAP, consentendo una collaborazione multicanale. Lo spazio condiviso coincide qui con le conferenze offerte dal forum.**

Altre applicazioni più complesse in termini collaborativi, come BSCW<sup>29</sup>, realizzano veri e proprie aree di lavoro (workspaces), in cui suddividere post e altre risorse, leggere e scrivere email, e ricevere notifica di altri utenti che sono attualmente collegati al sistema, o lo sono state in un dato lasso di tempo precedente.



**Fig. 7-4: un'applicazione WAP (BSCW) basata sul concetto dei workspace.**

Il concetto di workspace risulta particolarmente interessante, in quanto rappresenta, in termini di dati, uno spazio di lavoro condiviso entro il quale avviene la cooperazione. E' in termini dello workspace che la collaborazione viene definita: gli utenti si autenticano per accedervi, il sistema registra gli accessi e notifica altri utenti dell'ingresso nel workspace stesso. Il modo con cui rappresentare lo spazio condiviso sta al centro di una

<sup>29</sup> Wolfgang Appelt - WWW Based Collaboration with the BSCW System - in Proceedings of the International of SOFSEM '99, Springer Lecture Notes in Computer Science 1725, pages 66-78, Milovy (Czech Republic), November 26- December 4 1999

applicazione collaborativa, e la decisione di come rappresentarlo all'interno di un canale talmente limitato come il Wap, è probabilmente la decisione più difficile.

Lo scopo dell'implementazione proposta è quella di implementare dei sistemi che utilizzino collaborazione più complessa, modellati sui vincoli del canale Wap, e derivati dall'analisi degli strati di presentazione, navigazione, e dominio funzionale. La collaborazione supportata deve essere integrabile con quella realizzata attraverso canali più complessi, in modo da poter valutare l'effetto sulla collaborazione data dalla presenza di più canali dalle caratteristiche fortemente eterogenee.

## **7.2.2 Analisi del canale WAP**

### **7.2.2.1 Architettura fisica**

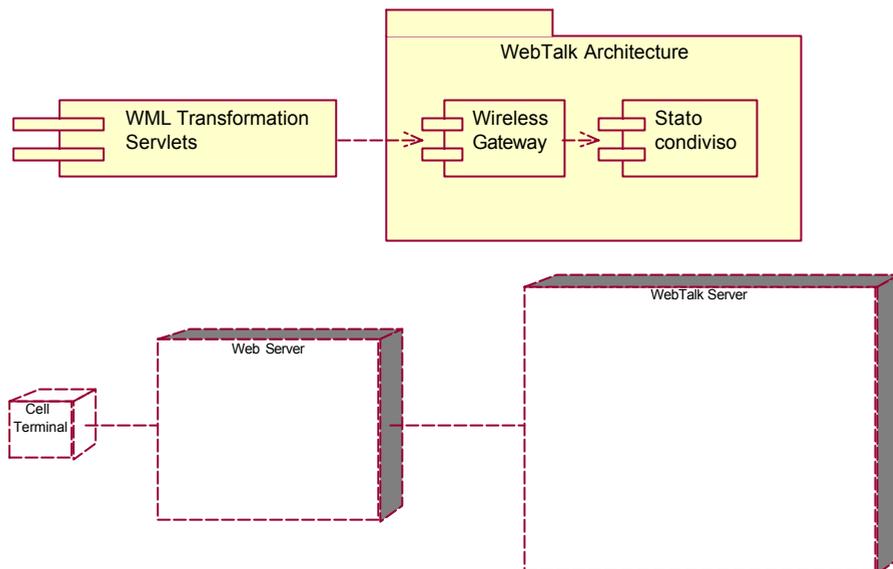
L'architettura di implementazione del canale WAP si fonda sull'aggancio all'architettura generale WebTalk da noi concepita.

In particolare, uno dei moduli di connessione montati sul lato server dell'architettura, preposto alla gestione tecnica dello stato condiviso, ha il compito di filtrare le PDU di aggiornamento dello stato condiviso e di colloquiare in modo più leggero e con un numero di entità pacchetto selezionato. Questo modulo, denominato Wireless Gateway, è stato studiato appositamente per consentire l'accesso alla gestione dello stato condiviso da parte di altri sottosistemi server, o direttamente da parte di altri client, i quali non abbiano la possibilità di gestire le informazioni di aggiornamento in termini di classi serializzate Java e di quaternioni di trasformazione, il formato prescelto per le PDU (Protocol Data Unit) trasmesse dal gestore di condivisione.

Il compito del Wireless Gateway, dunque, è di fornire una interfaccia di comunicazione più snella, che fornisca solamente i dati rilevanti per l'aggiornamento dello spazio condiviso del canale aggiuntivo che si aggancia al sistema, e, conversamente, di ricevere le PDU semplificate generate dai sottosistemi di comunicazione mobile e trasformarli in pacchetti adatti alla trasmissione verso client appartenenti ai canali più evoluti.

Nel caso particolare dell'uso del canale WAP, tuttavia, l'architettura risulta più complessa a causa dello stato dell'arte della tecnologia e delle sue attuali caratteristiche di implementazione. Il terminale su cui effettuare la presentazione all'utente nel canale WAP, infatti, è di natura estremamente leggera. Esso ha ridottissime capacità elaborative (se non nulle), disponendo per lo più solamente di un browser Wap in grado di interpretare card WML che entrano nel terminale mediante la connessione dati. Non vi sono inoltre possibilità di memorizzare dati, né temporanei, né tantomeno in modo permanente. Queste caratteristiche rendono impossibile un collegamento diretto al Wireless Gateway, e richiedono l'interposizione di uno strato intermedio che sia in grado di raccogliere puntualmente la versione "filtrata" dello stato condiviso offerta dal wireless gateway, trasformarla in WML ed offrirla al terminale richiedente.

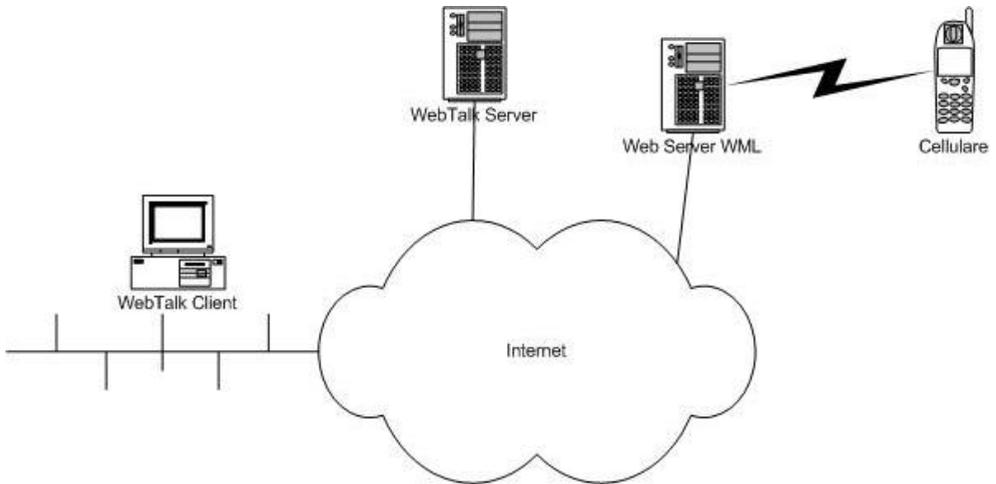
Per questo motivo si è interposto tra i terminali client ed il gateway al gestore dello stato condiviso un application server, implementato utilizzando un web server HTTP in grado di servire anche contenuti WML, ed un back-end applicativo che utilizza l'architettura J2EE, e formato essenzialmente da una collezione di servlet Java.



**Fig. 7-5: componenti dell'architettura dell'aggancio canale WAP: viene realizzato attraverso servlet di trasformazione WML su un application server, in grado di avere sia una connessione HTTP verso il terminale mobile che una connessione diretta TCP/IP verso il gestore dello stato condiviso filtrato (Wireless Gateway). Il terminale risulta collegato sempre al webservice e non direttamente al sistema.**

Le servlet si occupano anche di tenere in memoria una tabella contenente informazioni su tutti gli utenti; per ciascun utente viene memorizzato un identificatore univoco, il gruppo di appartenenza, la username utilizzata, la tabella dei messaggi ricevuti (periodicamente aggiornata) e l'eventuale file di testo ricevuto. Le servlet implementano anche la logica collaborativa specifica del canale WAP, essendo questo componente il primo componente specifico a valle del server WebTalk. La dinamica collaborativa specifica del canale Wap viene descritta durante la trattazione dello strato collaborativo di canale, a seguire.

A questo scopo il back-end J2EE mantiene una tabella di memorizzazione dei gruppi e, per ciascuno di essi, viene memorizzato il nome, il creatore, il canale di "provenienza" (WAP o WebTalk) e, nel caso di gruppi WAP, il tipo (pubblico o privato). Le servlet costruiscono poi le pagine WML da inviare agli utenti contenenti i risultati dell'elaborazione dei dati in risposta alle richieste pervenute dal client.



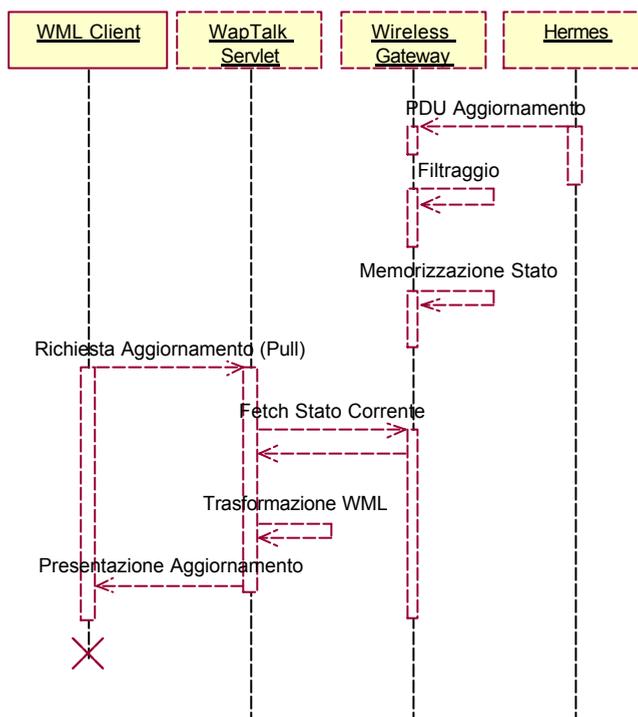
**Figura 7-6. Schema di comunicazione tra un utente Web ed un utente WAP, il primo attraverso il canale a rappresentazione 3D, ed il secondo attraverso il canale WAP.**

Un utente WAP, accedendo attraverso il canale Wap, è in grado di conoscere l'elenco di tutti gli utenti appartenenti allo stato condiviso, compresi quelli entrati mediante canali diversi o canali più complessi (es. il client tridimensionale), gestiti tutti dal server che si occupa del mantenimento dello stato condiviso. L'utente appartenente al canale Wap è in grado di sapere quale canale stanno utilizzando gli altri utenti, ed eventualmente di dedurre quali sono le implicazioni a livello funzionale e collaborativo nel momento in cui si interagisce con un utente di un canale diverso.

Le caratteristiche tecniche del canale Wap, allo stato del protocollo 1.1 e della successiva evoluzione all'1.2, impongono una ulteriore limitazione alle modalità di collaborazione possibile nel contesto multicanale. Nel protocollo 1.1, infatti, l'unica modalità di interazione con il server WML è la modalità Pull. Questo significa che il client deve ogni volta iniziare una richiesta verso il server per poter ricevere informazione. Mentre questo paradigma è sicuramente sufficiente per l'applicazione principale per cui WAP è stato pensato, e cioè la navigazione di card WML attraverso la rete radiomobile, questo aspetto limita pesantemente le possibilità di interazione con il gestore di stato condiviso della nostra architettura collaborativa.

Avendo a disposizione solamente la modalità Pull (la modalità Push in WAP 1.1 è disponibile solamente utilizzando come *bearer* SMS), infatti, non è possibile al gestore di stato condiviso notificare l'insorgere di eventi di aggiornamento in modalità Push, e cioè direttamente nel momento in cui essi avvengono: occorre invece aspettare che il terminale remoto faccia richiesta di essere aggiornato eseguendo un Pull dello stato condiviso semplificato presente nel Wireless Gateway e convertibile istantaneamente

attraverso l'Application Server J2EE interposto tra client e stato condiviso. Una soluzione potrebbe essere di comandare al terminale client una temporizzazione automatica degli eventi di Pull. Tuttavia non è possibile specificare in una card WML (come invece è possibile mediante l'utilizzo di JavaScript all'interno di browser HTML) un ricaricamento della card temporizzato a periodi.



**Fig. 7-7: Effetto della mancanza della modalità push nel protocollo WAP 1.1. Il canale WAP deve aggiornarsi periodicamente in modalità Pull per ottenere l'allineamento dello stato condiviso sull'istanza locale. Questo impatta enormemente sulla percezione collaborativa. Nella figura, lo strato Hermes si occupa di veicolare le entità PDU relative all'aggiornamento dello stato condiviso.**

Il protocollo WAP 1.2 introduce il concetto di Push anche su bearer non SMS, tuttavia con alcune limitazioni che non lo rendono ancora utilizzabile nel contesto collaborativo da noi progettato. In primis, la maggior parte dei terminali oggi disponibili implementano lo standard 1.1 e non 1.2 (che comincia ad apparire sui nuovi modelli GPRS) ed i gateway WAP a disposizione non dispongono di funzioni push (perché, allo stesso modo, implementano lo standard 1.1). In ogni caso, il push implementato su WAP 1.2 è un sistema di broadcasting per le utenze di un singolo gateway, il che significa che i messaggi push possono essere inviati solamente direttamente dal gestore che fornisce il gateway Internet cui si è collegati. L'unica modalità di distribuzione è

inoltre quella broadcast, e quindi il messaggio push viene inviato indistintamente a tutti gli utenti collegati al gateway di distribuzione di accesso alla rete.

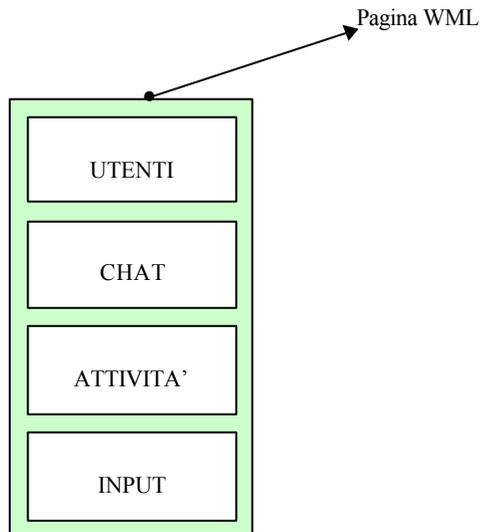
#### 7.2.2.2 Strato di Presentazione

Lo strato di presentazione del canale deve stabilire il modo con cui rappresentare lo stato condiviso, oggetto della collaborazione, nel rispetto dei vincoli fisici discussi precedentemente. Nella discussione della rappresentazione tridimensionale, si è discusso come l'utilizzo del 3D consentisse di utilizzare delle visualizzazioni in grado di aumentare l'acuità di percezione dello stato condiviso da parte dei partecipanti.

Non è naturalmente possibile proporre, all'interno del canale WAP, una rappresentazione tridimensionale, e nemmeno alcuna rappresentazione grafica dello stato condiviso. Ciò che risulta possibile fare, tuttavia, è scegliere alcuni dei principali dati componenti lo stato condiviso e fornire una interfaccia testuale che li descriva.

La rappresentazione dello stato condiviso deve suscitare nell'utente la sensazione di trovarsi in un ambiente "popolato" da altri utenti, condiviso con essi e nel quale sono disponibili gli strumenti necessari per la cooperazione.

Lo stato condiviso nel canale WAP viene rappresentato come un'area (una pagina WML) divisa in quattro sotto-aree principali:



**Figura 7-8. Schematizzazione dello spazio condiviso per il canale WAP.**

Ciascuna delle aree tematiche raccoglie alcuni dati relativi allo stato condiviso dell'applicazione:

- l'area *utenti* comprende una lista dei partecipanti allo stato condiviso, specificando il canale di appartenenza
- l'area di *chat* comprende la comunicazione testuale che si sta svolgendo all'interno dello spazio condiviso, normata secondo le regole collaborative specifiche del canale
- l'area *attività* comprende la possibilità di scegliere le operazioni rese possibili dal dominio funzionale dell'applicazione
- l'area *input* consente di inserire dati all'interno dello spazio condiviso per aggiornarlo o interagire con esso

Lo spazio condiviso viene aggiornato ad ogni ciclo di Pull delle informazioni (mediante l'esplicita scelta della funzione di aggiornamento). L'aggiornamento comprende la modifica dei contenuti dell'area utente e dell'area di chat, o eventualmente il procedere delle funzionalità di navigazioni che saranno descritte nello strato navigazionale.

Abbandonando la card che rappresenta lo stato condiviso, si abbandona la rappresentazione dell'ambiente collaborativo, e pertanto questo equivale ad uscire dal sistema collaborativo, per eseguire altre attività con il proprio terminale.

Lo strato di presentazione non offre, per scelta, alcuna informazione riguardo allo stato degli oggetti condivisi, né alcuna informazione sulla geometria del mondo esplorato. Si è cercato di fornire una rappresentazione il più compatta possibile, e questo ha comportato una diminuzione del dominio funzionale messo a disposizione.

### **7.2.2.3 Strato di Navigazione**

Il canale WAP prevede che la maggior parte dell'interazione avvenga nella singola card contenente lo stato condiviso. Tuttavia è possibile navigare verso il menu che offre le opzioni, ed impostare le opzioni desiderate.

Un'altra opportunità che consente di lasciare la rappresentazione canonica dello stato condiviso è la navigazione di altri contenuti in modalità libera o condivisa. L'attività di navigazione è finalizzata all'esplorazione di altri contenuti in formato WML, eventualmente presente su altri siti. Le due modalità si riferiscono alla possibilità di farlo in modo collaborativo o meno. La navigazione di queste funzionalità avviene in modo strettamente sequenziale, utilizzando i tasti di navigazione messi a disposizione dal terminale.

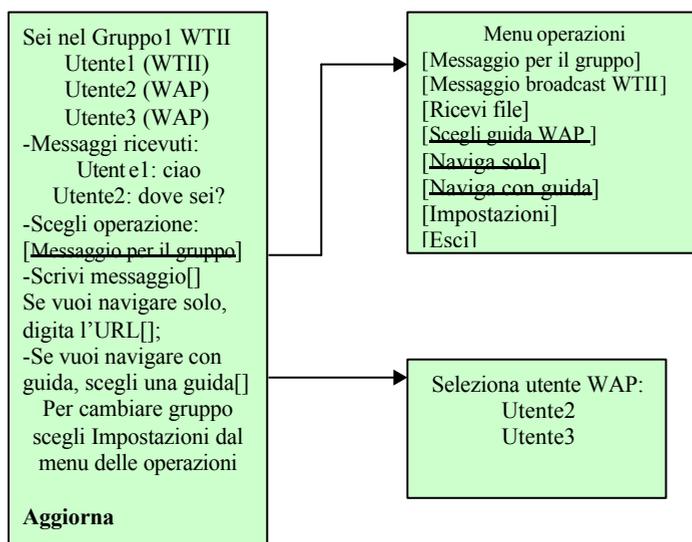
### **7.2.2.4 Strato Funzionale**

Le funzioni del sistema collaborativo WebTalk a cui è possibile accedere mediante il canale WAP sono le seguenti:

- Gestione dell'elenco utenti presenti nello spazio collaborativo con specifica del canale di appartenenza

- Gestione delle metafore di collaborazione fornite (attivazione e disattivazione gruppi, ecc.)
- Navigazione WAP, fruibile mediante metafore collaborative proprie del canale
- Scambio messaggi: attraverso una chat testuale, due o più utenti possono comunicare fra loro; le modalità di scambio dei messaggi sono stabilite dalle metafore cooperative scelte.
- Scambio File: gli utenti di canali dotati di memorie a massa possono condividere dei file con gli utenti del canale WAP, inviandoli per la lettura all'utente WAP. In questo modo la collaborazione consente all'utente WAP, più "debole" in termini di potenzialità di chiedere aiuto nella ricerca di una risorsa ad un utente di un canale con maggiori disponibilità, realizzando una vera e propria collaborazione.

L'organizzazione delle funzioni all'interno della presentazione dello stato condiviso avviene secondo lo schema seguente. Le voci delle funzionalità offerte sono state estese per consentirne una migliore comprensione, mentre nella presentazione vera e propria sono offerte in forma abbreviata per occupare meno spazio possibile sul display.



**Fig. 7-9: schema delle funzionalità offerte dal canale WAP per la condivisione in ambiente WebTalk**

Dal dominio funzionale è stata rimossa qualsiasi funzione riguardante la gestione dello stato condiviso degli oggetti, poiché si è ritenuto che il canale fosse troppo limitato per poter fornire informazioni effettivamente fruibili al riguardo.

### 7.2.2.5 Strato Collaborativo

Gli elementi di collaborazione previsti per il canale WAP sono sostanzialmente due. Uno di questi, il gruppo, è adottato e leggermente rivisitato prendendo spunto dagli elementi collaborativi generali, nella configurazione base del modello di collaborazione trattato precedentemente. Il secondo, l'elemento di navigazione, è un nuovo elemento introdotto per descrivere le dinamiche di navigazione proprie del canale WAP.

Altri elementi collaborativi sono la comunicazione testuale, che segue regole gruppali, e lo scambio di informazioni testuali via file.

#### *Gruppo*

Gli utenti all'interno del sistema sono separati in gruppi. Un utente che accede al sistema deve necessariamente scegliere un gruppo nel quale svolgere le attività. In una metafora collaborativa di esplorazione a gruppi vincolata, la cooperazione è consentita solo fra utenti di uno stesso gruppo e non fra utenti di gruppi diversi. Questa scelta non è strettamente necessaria (è tecnicamente possibile permettere la cooperazione fra tutti gli utenti del sistema), ma consentire un meccanismo di raggruppamento permette delle funzionalità aggiuntive dal punto di vista applicativo: a seconda di come viene implementato ed utilizzato nello specifico contesto, tale meccanismo consente la differenziazione e la caratterizzazione della cooperazione tra gli utenti. Sono possibili vari livelli di configurazione, tuttavia si è scelta un'unica forma della metafora che utilizzano l'elemento cooperativo del gruppo: un utente appartenente in un gruppo può inviare messaggi solo a partecipanti del medesimo gruppo, e ricevere messaggi solo da partecipanti a quel gruppo. Non si sono volute offrire un numero maggiore di opzioni di configurazione, a causa della difficoltà della loro impostazione sul terminale limitato.

Si sono tuttavia fornite due tipologie di gruppi: pubblico e privato; la partecipazione ad un gruppo è regolamentata da alcune norme: l'accesso ad un gruppo richiede un'abilitazione se il gruppo è privato; i gruppi pubblici sono invece ad accesso libero. Questa abilitazione viene fornita a livello di configurazione delle servlet J2EE che regolamentano lo stato condiviso per il canale WAP, da parte di un amministratore di sistema.

L'esistenza di gruppi ad accesso privato consente una forma di collaborazione più stretta; in altre parole, è possibile che gli utenti scelgano con chi collaborare. Sono autorizzati a partecipare al gruppo privato solo gli utenti il cui username risulta nell'elenco degli aventi diritto.

La creazione di gruppi è libera ma è possibile crearne solo di tipo pubblico; i gruppi privati sono inseriti nel sistema dall'amministratore, che provvede a registrare l'elenco degli autorizzati all'accesso.

Esiste un gruppo predeterminato, in modo che il primo utente che accede al sistema non sia obbligato a crearne uno. La creazione di un nuovo gruppo avviene semplicemente richiedendo l'apertura di un gruppo; creando un gruppo si entra a far parte di esso come primi utenti. Il gruppo pubblico si distrugge quando anche l'ultimo utente rimasto lo abbandona; il gruppo privato può essere distrutto solo dall'amministratore. Queste

scelte alleggeriscono notevolmente la gestione dei dati relativi a questi meccanismi e sono, quindi, in linea con la decisione di realizzare un sistema leggero e semplice. Infine, esiste una regola di partecipazione che prevede che si possa appartenere ad un solo gruppo.

### ***Partecipazione ai gruppi***

La gestione tecnica dello stato condiviso avviene separatamente per ciascun mondo. Un server può eventualmente gestire più stati, ciascuno legato ad un mondo (o applicazione) diversi. All'interno di ciascun mondo, gli utenti si possono (liberamente o meno) organizzare in gruppi. Un utente WAP può vedere i gruppi di un mondo ed eventualmente scegliere di partecipare ad uno di essi. L'utilizzo del canale WAP modifica l'organizzazione dei gruppi, aggiungendo l'informazione relativa a quale utente crea inizialmente un gruppo: i gruppi WAP sono gruppi inizialmente creati da un utente appartenente al canale WAP, gli altri gruppi sono invece gruppi di tipo "tradizionale". La differenziazione del canale di origine del gruppo si rende necessaria a causa del diverso dominio funzionale dei due canali. Qualora un utente di un gruppo non WAP desideri collaborare con un utente WAP, dovrà entrare all'interno di un gruppo di tipo WAP, esplicitamente restringendo il proprio dominio funzionale verso quello dell'altro canale.

### ***Scambio messaggi***

Se un utente WAP partecipa ad un gruppo WebTalk, è previsto che lo scambio dei messaggi avvenga tra i soli partecipanti al gruppo.

Se invece l'utente WAP non sceglie un gruppo WebTalk ma un gruppo WAP, egli può comunque effettuare una cooperazione multicanale, ma questa non sarà circoscritta ad un gruppo ristretto di utenti. E' stata prevista, infatti, una modalità *Broadcast* di invio dei messaggi, che, se adottata, provvede a recapitare il contenuto di ciascun messaggio a tutti gli utenti che si trovano all'interno del mondo WebTalk.

### ***File di testo***

Un utente WAP può ricevere file di testo da un client di WebTalk; i file possono essere spediti da un utente WebTalk ad un utente WAP, per cui si è previsto che quest'operazione avvenga in forma personale e non è soggetta alla metafora di gruppo. La condivisione dei documenti è stata pensata principalmente come un'attività cooperativa più stretta; questa modalità di collaborazione offre la possibilità di condividere risorse informative (informazioni culturali, documenti informativi di vario genere, ecc..) sulle quali è possibile discutere, lavorare in modo collaborativo, scambiando opinioni, operando in modo coordinato.

Il file di testo è un oggetto virtuale che può essere condiviso attraverso il canale WAP. Un utente WebTalk, come spiegato, partecipa alle attività collaborative grazie ad un client installato sul proprio PC. L'utente WebTalk può decidere di prelevare il file dal proprio PC e di inviarlo attraverso la finestra grafica ad un utente WAP. E' possibile

spedire un file ad un singolo utente o ad un gruppo di utenti; le due soluzioni possono coesistere o essere esclusive.

### ***Navigazione libera***

Nella modalità di navigazione libera, un utente che utilizza il canale WAP può decidere, digitando un URL, di visitare pagine WML navigando autonomamente con il proprio browser WAP.

Al momento della digitazione dell'URL, tuttavia, è possibile vedere gli altri utenti dello spazio collaborativo, che utilizzano il canale Wap e che stanno eventualmente visitando la stessa pagina o che lo hanno fatto recentemente. All'interno di un sistema cooperativo questo tipo di informazione può risultare utile ad esempio in contesti applicativi che necessitano della ricerca di informazioni: può risultare vantaggioso sapere che altre persone hanno già "visitato" uno spazio informativo cui si ha intenzione di accedere. La navigazione non comporta l'uscita dal sistema; pur abbandonando temporaneamente lo spazio condiviso, l'utente è considerato ancora attivo, entro un certo limite temporale. E' possibile dunque continuare a ricevere i messaggi degli utenti facenti parte del gruppo e vederli aggiornando la propria rappresentazione dello stato condiviso quando lo si ritiene opportuno.

### ***Navigazione condivisa***

All'interno del proprio gruppo è possibile scegliere uno degli utenti del gruppo come guida nella navigazione: questo consente di vedere la pagina che l'utente scelto sta visitando in quel momento.

Il vantaggio che deriva dall'utilizzo di una metafora di questo tipo rientra nella prospettiva generale di condivisione dei contenuti a disposizione delle utenze in un sistema di reti: utenti fisicamente lontani hanno la possibilità di esplorare il mondo WAP in modo congiunto e concordato. Altri canali adottano scelte analoghe per realizzare la navigazione condivisa, adottando la metafora di guida. Quando questa metafora è utilizzata, solitamente è un utente che decide di diventare la guida di un gruppo; mentre gli altri utenti possono scegliere di conseguenza una modalità di navigazione "passiva", adottando il punto di vista della guida e lasciandosi trasportare all'interno dello spazio condiviso. Nel canale WAP, questa modalità collaborativa è stata modificata, poiché è l'utente che desidera essere guidato a scegliere l'utente da seguire, senza un esplicito consenso o nozione di questo fatto da parte della guida; questa soluzione risulta più intrusiva rispetto alla metafora guida più classica, e potrebbe essere eventualmente migliorata fornendo dei meccanismi di protezione di accesso o di concessione di un permesso ad agganciarsi in fase di navigazione. Tuttavia, questo tipo di collaborazione risulta interessante, dato che il WAP è principalmente pensato come un canale per il reperimento di informazioni. Rispetto al Web, in un utilizzo analogo, l'utente è meno invogliato all'esplorazione, essendo questa più difficoltosa a causa della natura del terminale utilizzato; questo comporta che molte risorse sul WAP, pur esistendo, rimangano nascoste. Se però si riesce a sfruttare la presenza di più utenti, si può fare in modo che essi collaborino al fine di conoscere ed esplorare i siti WAP disponibili: un utente che può vedere gli utenti collegati alla stessa

pagina da lui visitata, può trovare utile scegliere di seguire uno di loro. Questo gli offre la possibilità di trovare risorse a cui da solo non sarebbe arrivato.

### 7.2.3 Esempio applicativo

Si riporta nel seguito una sessione di collaborazione all'interno dell'applicazione di test "Leonardo Virtuale", di cui è disponibile il canale collaborativo tridimensionale, mediante il server WebTalk. Nella sperimentazione, si effettua una navigazione WAP in collaborazione con un altro utente, presente su un canale differente, quello tridimensionale, allo scopo di valutare gli effetti della collaborazione.

#### *Accesso al sistema*

Digitando un URL dal proprio browser WAP, l'utente accede alla pagina iniziale dell'applicazione:



**Figura 7-10. Card di apertura.**

Per ottenere l'accesso al sistema è necessario inserire uno username:



**Figura 7-11. Card di accesso al sistema: l'utente è obbligato ad inserire un nome che lo identifichi all'interno del sistema.**

Una volta eseguito l'ingresso nel sistema, lo stato condiviso viene aggiornato su tutti gli utenti partecipanti alla sessione, indipendentemente dal canale di appartenenza, e tutti percepiscono l'ingresso del nuovo utente attraverso il canale WAP.

### ***I gruppi***

Una volta inviata la conferma dell'avvenuta registrazione, all'utente viene chiesto di creare un gruppo o di inserirsi in un gruppo già esistente di utenti, semplicemente selezionando una tra le opzioni proposte nelle schermate di creazione gruppi:



**Figura 7-12.** I menu di scelta adottano una forma sintetica basata sulle parole chiave; in questo modo si evita all'utente l'azione di scrolling verso destra, a vantaggio di una navigazione più scorrevole.

### ***Scelta di un gruppo***

Se l'utente desidera far parte di un gruppo già esistente, dovrà selezionare il gruppo scelto dall'elenco dei gruppi esistenti:



**Figura 7-13.** L'elenco dei gruppi disponibili propone il gruppo di default (il WAP) ed i gruppi creati dagli utenti che sono già all'interno del sistema.

L'utente può scegliere di inserirsi in un gruppo già esistente, ad esempio nel gruppo predefinito "WAP"; si tratta di un *default* di sistema, il cui scopo è raccogliere tutti gli utenti che non esigono un particolare gruppo, o come scelta nel caso non vi siano gruppi creati da altri utenti:



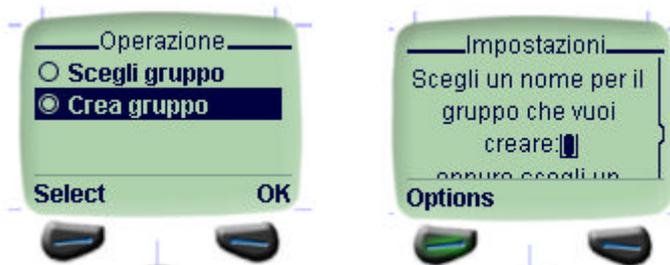
**Figura 7-14.** Questa card rappresenta la parte superiore dello spazio condiviso, in cui gli utenti di un gruppo si vedono e possono scegliere le operazioni da effettuare.

Una volta scelto il gruppo, si accede allo “spazio condiviso”, la rappresentazione nel canale WAP dello spazio di collaborazione. Al suo interno si incontrano i quattro sottogruppi relativi agli utenti, alla comunicazione, alle funzioni, e all’input.

#### *Creazione di un gruppo*

Qualora l’utente scelga invece di creare un nuovo gruppo, è possibile utilizzare l’opzione “Creazione Gruppo” e inserire il relativo nome.

L’utente che richiede di creare un gruppo è chiamato il creatore del gruppo. Il gruppo risulterà di tipologia WAP, e automaticamente di tipo pubblico.



**Figura 7-15.** Il menu nella prima delle due immagini è raggiungibile sempre dalla card principale, selezionando l’operazione “Impostazioni” dall’elenco delle operazioni; nella seconda immagine si vede la finestra per l’inserimento del nome del gruppo che si vuole creare.

#### *La collaborazione*

Una delle forme di collaborazione possibile è lo scambio di messaggi, secondo una delle metafore cooperative attive nel canale: selezionando l’opzione “messaggio per il Gruppo” dal menu principale, il messaggio sarà consegnato a tutti gli utenti del gruppo di appartenenza:



Figura 7-16. Il menu delle operazioni disponibili; scorrendolo sono visibili le altre scelte.

Al momento dell'avvenuta registrazione dell'utente, il sistema ha già provveduto ad aprire il canale di comunicazione con il Web agganciandosi al server di WebTalk. Le altre funzioni consentono di mandare un messaggio broadcast, oppure di iniziare una sessione di navigazione libera, o guidata.

### Scambio File

Lo scambio file prevede che l'utente dal canale fisso invii all'utente con canale mobile il file da condividere, ad esempio mediante un comando di menu o un'attivazione drag and drop (al momento non implementata).

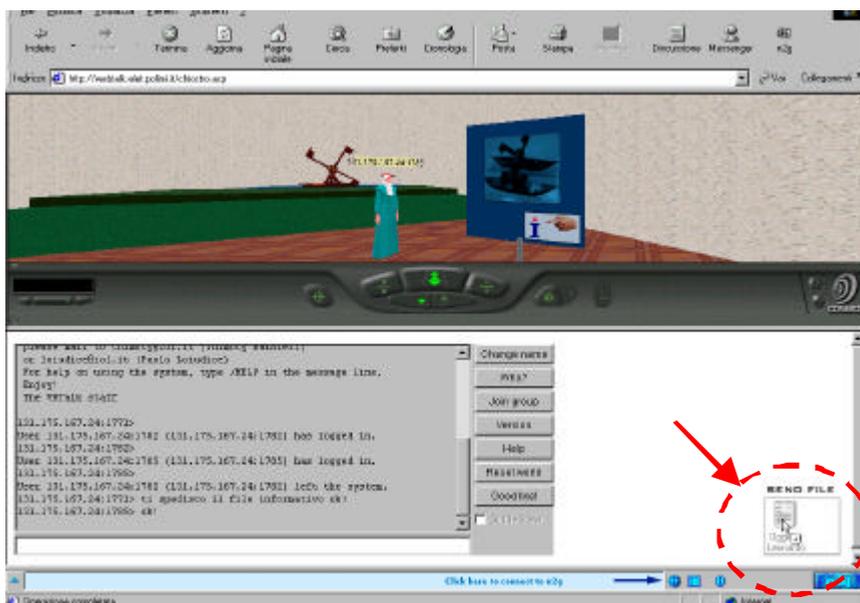


Figura 7-17. Ipotesi sull'eventuale soluzione grafica per l'invio del file.

Se, ad esempio, l'utente ha bisogno di informazioni su una macchina di Leonardo, come il paracadute, durante la visita collaborativa, è possibile inviare al suo cellulare il file. Sul lato del terminale WAP, l'utente dovrà eseguire il prelievo dell'informazione, selezionando l'operazione "Ricevi File" dal menu.



**Figura 7-18.** Scorrendo la pagina si può leggere il contenuto del file.

#### *Navigazione Guidata e Navigazione Libera*

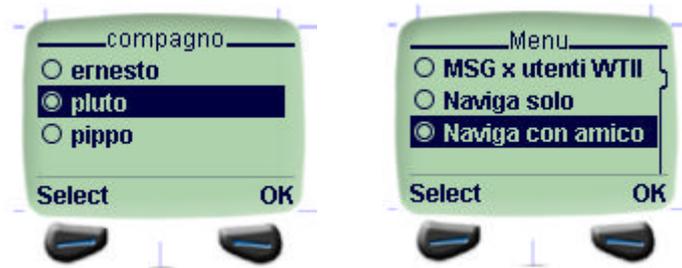
Se fra gli utenti WAP partecipanti al gruppo dell'utente durante la collaborazione ne esiste uno che conosce un sito WAP dove reperire materiale utile, può comunicarlo attraverso un messaggio: l'utente farà in modo di ricevere le pagine scegliendo il compagno di gruppo come guida (si indica quest'ultimo utente come "guida"):



**Figura 7-19.** Uno scambio di messaggi fra due utenti di un gruppo.

L'ultimo messaggio inviato nella comunicazione è il primo ad essere visualizzato; questa soluzione evita all'utente lo scorrimento della pagina ad ogni aggiornamento.

Scegliendo la modalità di navigazione guidata, si possono ricevere le pagine caricate dall'utente con cui si attiva questa modalità di collaborazione:



**Figura 7-20.** Scelta della guida per la navigazione e dell'operazione corrispondente.

La guida dovrà effettuare una navigazione libera, mentre l'utente che vuole essere guidato si aggancia alla sua visualizzazione WAP.



**Figura 7-21.** Selezione dell'operazione di navigazione solitaria.

La navigazione libera avviene digitando l'URL desiderato.



**Figura 7-22.** Finestra per l'inserimento dell'URL.

La guida ha la possibilità di sapere quali utenti sono su quel sito in quel momento ed eventualmente effettuare una navigazione condivisa di pagine WML. Inviando al server la relativa richiesta, viene fornito l'elenco degli utenti collegati alla navigazione sulla medesima pagina:



**Figura 7-23.** Dopo l'immissione dell'URL, inviando il dato si riceve l'elenco degli utenti collegati allo stesso URL.

Questa funzionalità è offerta sempre nell'ottica di stimolare gli utenti all'attività cooperativa: se si sta consultando la stessa risorsa, è probabile che possa essere necessario una comunicazione, o innescare una collaborazione.

#### 7.2.4 Considerazioni sui Risultati Ottenuti

Le sessioni di collaborazione multicanale tra canale con visualizzazione 3D e canale WAP si sono svolte in laboratorio, utilizzando le funzionalità messe a disposizione dal sistema.

Un primo esperimento di navigazione riguarda la collaborazione tra utenti tutti attraverso il canale WAP. La navigazione è risultata difficoltosa, e gli utenti non hanno percepito in modo netto l'esistenza di uno spazio condiviso se non dopo numerose spiegazioni. Inoltre la navigazione, impacciata dall'impossibilità di utilizzare notifiche di tipo push, risulta molto lunga e farraginosa, dovendo costringere l'utente a aggiornare continuamente il proprio stato prima di intraprendere l'azione successiva. La dilatazione dei tempi di visualizzazione, e la difficoltà di interazione con un'applicazione complessa utilizzando mezzi di input inadeguato, hanno contribuito ad allontanare la percezione di un ambito collaborativo.

E' tuttavia risultata subito chiara la presenza di un punto di concentrazione (il server dello stato condiviso), in cui erano registrati più utenti, essendo la porzione di stato condiviso relativa agli utenti la più evidente all'interno dell'applicazione.

Anche l'utilizzo della navigazione guidata è risultata piuttosto lenta e difficoltosa.

Un secondo esperimento ha riguardato la collaborazione tra utenti che utilizzavano il canale tridimensionale e utenti con il canale WAP. Entrambe le categorie di utenti hanno percepito senza nessun problema la presenza reciproca, tuttavia l'interazione inter-canale è risultata molto slegata, a causa dell'enorme sbilanciamento in termini di prestazioni e di funzioni tra i due canali. Sostanzialmente la velocità di operazione sul canale tridimensionale non è paragonabile con quella possibile sul canale WAP,

pertanto le attività nella collaborazione sono fortemente sbilanciate sul lato 3D, costringendo gli utenti WAP ad assumere un atteggiamento passivo.

I risultati dunque parlano chiaramente: non è possibile pensare, almeno con la limitatezza attuale del canale WAP, di simulare una collaborazione bilanciata tra questi due particolari canali, cercando di implementare una 'trasparenza funzionale' rispetto alle caratteristiche del canale stesso.

Tuttavia occorre tenere presente che difficilmente, anche avendone la possibilità, un utente di canale WAP avrebbe il desiderio di lavorare in modo particolarmente attivo. Gli utenti WAP, infatti, si trovano nella maggior parte dei casi in situazione di estrema mobilità (in piedi, camminando, guidando), e quindi con possibilità minima di interagire con il terminale, a cui viene dedicata una attenzione discontinua.

Che dunque gli utenti del canale WAP assumano un ruolo collaborativo decisamente passivo non risulta del tutto scoraggiante, ma va considerato in linea con la natura di utilizzo del canale. Il canale WAP va considerato funzionalmente per quello che realmente può dare: può servire come mezzo per monitorare rapidamente lo stato di un dato ambiente collaborativo, ad esempio per sapere in tempo reale quali sono gli utenti presenti all'interno dello spazio condiviso e quale è la conversazione in atto.

Queste due funzioni di monitoraggio sono brillantemente assolte dal WAP. Si può dunque presumere che un canale del genere serva solo per il controllo di presenza o disponibilità di altri utenti, delegando poi all'uso di altri canali l'effettuazione di collaborazioni più complesse. Per fare un esempio pratico, un utente potrebbe utilizzare il canale WAP per controllare se alcuni determinati altri utenti sono disponibili in linea per una collaborazione, e, in caso positivo, affrettarsi verso una postazione più potente per cominciare una sessione collaborativa urgente.

Riteniamo tuttavia di estrema importanza l'esperienza, perché ha consentito di trovare un metodo per classificare ed analizzare ciascun canale in termini delle sue caratteristiche, e ha consentito di formulare una valutazione concreta sull'effettiva possibilità di collaborare in contesti multicanale.

Dall'esperienza del canale WAP è risultato chiaro come la possibilità di accedere a possibilità elaborative e di presentazione più complesse potesse rendere maggiormente interessante l'utilizzo di un canale mobile per l'accesso collaborativo. La disponibilità di computer palmari collegabili a reti radiomobili, dunque, rappresenta il naturale passo evolutivo nell'esplorazione della collaborazione multicanale in contesto mobile.

### **7.3 Utilizzo di dispositivi palmari**

I cellulari sono i terminali per l'accesso alla rete potenzialmente più disponibili, ma, al tempo stesso, sono anche le periferiche con le maggiori difficoltà di visualizzazione delle pagine e di interazione con l'utente.

A questi problemi può rispondere un computer palmare, capace di collegarsi alla Rete con un qualunque cellulare (e quindi in wireless), ma anche in grado di sincronizzarsi con i propri siti preferiti attraverso l'apposito modulo, sfruttando così la velocità di connessione del modem a cui è collegato il computer (e quindi Isdn, Adsl, fibra ottica, linea dedicata ecc.). A questo vantaggio, si somma anche quello legato alle dimensioni dello schermo: un palmare ha capacità di visualizzazione nettamente superiori a quelle di un odierno telefono cellulare, inoltre la capacità di calcolo e di memoria rendono questo dispositivo decisamente più appetibile per la navigazione Internet ed in generale per la fruizione di contenuti.

Uno dei primi esempi di piattaforma palmare è stato il Pilot, basato sul microprocessore Motorola Dragonball (derivato dalla famosa serie 68000), Us Robotics (poi 3Com e Palm). Pilot è stato proposto nel '96 e si è sviluppato inizialmente in sordina, ponendo poi le basi del successivo trionfo e diventando uno dei sistemi operativi per computer palmari ancora oggi più diffusi al mondo, con migliaia di applicazioni di ogni genere disponibili. Le sue peculiarità principali erano le dimensioni (12 x 8 x 1,5 cm), il peso (160 g), un look accattivante, un prezzo non elevato ed il sistema Graffiti per il riconoscimento della scrittura manuale, semplice da apprendere ed estremamente preciso ed efficace. Pur avendo quindi le sembianze di una calcolatrice, con uno schermo LCD a toni di grigio ed una risoluzione di 160 x 160 punti, la flessibilità e stabilità del sistema operativo e la grande semplicità e linearità d'uso lo hanno rapidamente fatto apprezzare sia ai professionisti che agli appassionati del settore, in particolare negli Stati Uniti. L'alternativa era (e ancora oggi è) Psion, che si è imposta anch'essa nel '96 con i PDA appartenenti alle serie 3 e Siena. Il 3c aveva fino a 2 B di RAM, uno schermo LCD con risoluzione di 480 x 160 punti, uno slot di espansione e batterie alcaline di tipo AA, mentre il Siena arrivava fino ad 1 MB di RAM, con schermo LCD avente risoluzione 240 x 160 punti e batterie di tipo AAA e si caratterizzava per la sua tascabilità. I prodotti della casa inglese hanno storicamente avuto un buon riscontro (oggi ridimensionatosi), in particolare in Europa, per due motivi essenziali: integrazione di tastiere ergonomiche di ottima qualità (soprattutto con la fortunatissima serie 5) e sistema operativo EPOC multitasking robusto e flessibile.

Un altro concorrente nel panorama dei palmari è Windows CE. Dopo il rilascio di Windows 95, Microsoft si è resa conto della crescente importanza del mondo dei palmari ed ha così sviluppato un sistema operativo "ridotto" che potesse acquisire una consistente fetta di mercato. I primi costruttori ad installare Windows CE sono stati Hewlett-Packard (proveniente dall'esperienza positiva vissuta con OmniGo, una sorta di handheld computer), Compaq, Nec, Philips e Casio. Inizialmente sono stati proposti modelli handheld, basati cioè su una tastiera meccanica integrata nel dispositivo, generalmente abbastanza confortevole e su uno schermo LCD (per la prima volta a colori) con una risoluzione considerevole. In seguito sono poi stati proposti modelli palmtop, dei veri e propri computer palmari, con schermi sia a colori che a toni di grigio operanti alla risoluzione di 320 x 240 punti. Sulla scia di Windows 95, Microsoft ha cercato di proporre un sistema che si caratterizzasse per la sua multimedialità, in grado perciò di gestire suoni (anche in formato MP3), animazioni e grafica in genere. Per garantire simili prestazioni, CPU a parte, l'hardware deve ovviamente essere piuttosto spinto, sia per quanto riguarda la sezione audio (stereofonica) che quella video (elevato

frame rate), come nella serie iPAQ di Compaq, nato per fondersi al meglio con l'ultima versione Pocket PC. Ma non è finita e l'immaginazione è tanta. La dominanza e la differenza tra i principali sistemi operativi prevalenti nel mercato attuale come Windows CE, Pocket PC e Symbian (EPOC), ed allo stesso tempo dei processori impiegati (SH3, MIPS, RISC, EPOC ecc) fa degli ultimi modelli dei computer veri e propri.

In questo quadro si inserisce il Palm Vx, evoluzione del Palm serie 5; anche se ormai la sua uscita risale al 1999, resta uno dei punti di riferimento per lo sviluppo di applicazioni per palmari. Il Palm Vx è l'ultimo prodotto che precede l'ultima innovazione in fatto di palmari: l'integrazione wireless; si tratta di fatto dell'ultima evoluzione della famiglia di palmari "ultraleggeri" nata dal PalmIII e proseguita con il Palm V, Palm m100 e Palm IIIc.

Per quanto riguarda le performance di calcolo, costituisce un buon termine di paragone ponendosi nella fascia alta dei dispositivi "economici" probabilmente candidati migliori per la grande diffusione.

- CPU: Motorola Dragonball a 20 Mhz
- Sistema operativo: Palm OS 3.5
- ROM: 2 MB
- Flash rom aggiornabile
- RAM: 8 MB (4 volte maggiore del predecessore Palm V)
- Dimensioni: 11.42 x 7.87 x 1.01 cm
- Peso: 113.39 g
- Colore: alluminio anodizzato (color titanio)
- Schermo: LCD passivo a 16 toni di grigio
- Risoluzione: 160 x 160 pixels
- Retroilluminazione
- Porta IrDA
- Connessione seriale
- Batterie: al litio ricaricabili; durata 1 mese
- Alimentazione: AC attraverso cradle
- Cradle: seriale o USB



Rispetto ai cellulari WAP, i palmari forniscono generalmente performance superiori sia in termini di calcolo che di interazione:

	cellulare WAP	Palmare (Palm)
Potenza di calcolo	Bassissima	Bassa (20 MHz)
Display	b/n o colori	b/n o colori 160 x 160
Grafica	Testo + piccole immagini	Completo supporto grafico
Banda	14.4 Kb/sec	Secondo la connessione
Interfaccia	Pad numerico + hotkey	Hotkey + penna

Appare pertanto evidente come l'adozione di un Palm Vx, se da un lato pone delle severe limitazioni rispetto ad una piattaforma convenzionale come un PC, dall'altro apra delle prospettive migliori rispetto ad un semplice cellulare.

Ci troviamo quindi di fronte a tre differenti tipi di client, a cui corrispondono tre differenti livelli di supporto all'interazione:

Tipologia di client	Supporto alla visualizzazione	Supporto all'interazione	Livello di connettività
PC o equivalente	Rendering 3D VRML Alta risoluzione Visualizzazione testo Elevata profondità cromatica	Punta e clicca Inserimento testo Interfacciamento audio	Secondo la connessione modem, ISDN, ADSL, T1, T3
Palm	Misto testo/grafica 2D Bassa risoluzione Dimensioni ridotte del display	Punta e clicca Tasti funzione programmabili (tipicamente 6)	Secondo la connessione, fino a 57Kb/sec (seriale al PC)
Terminale WAP	Solo testo	Tasti telefonici	limitatissimo

Solitamente i dispositivi di bassa potenza come i palmari vengono programmati direttamente in codice nativo; per quanto riguarda lo sviluppatore questo si traduce il più delle volte nello sviluppare l'applicazione in C o C++ ed utilizzare un apposito package di deploy per ottenere un programma direttamente eseguibile sullo strumento.

A questa categoria di tool di sviluppo appartengono Codewarrior e Visual Basic for Palm OS.

Entrambi i tool permettono di realizzare applicazioni funzionanti, ma sono commerciali (cioè a pagamento) ed utilizzano sistemi proprietari per la realizzazione del software.

Una alternativa alla generazione di codice nativo, e per altro in linea con le tendenze attuali, consiste nell'utilizzo di una qualche sorta di Virtual Machine.

In questa ottica la scelta più ovvia è di realizzare il client in Java: il linguaggio e la tecnologia di gran lunga più adatti per lo sviluppo di applicazioni per Internet ed in cui è stato sviluppato il sistema WebTalk sin dai suoi primi passi.

Purtroppo anche in campo di JVM per palmare, alla data in cui scriviamo il mercato è ben lontano dalla definizione di uno standard: diversi produttori hanno proposto porting per palmare della classica JVM senza che il mondo degli sviluppatori abbia decretato il successo dell'una o dell'altra proposta.

Sovente le specifiche delle diverse VM sono fondamentalmente incompatibili tra di loro rendendo difficile immaginare di realizzare applicazioni portabili.

Per rendersi conto della ricchezza dell'offerta riporto una breve lista delle principali VM disponibili per i vari palmari, sia Palm che non:

Sistema operativo	JVM	librerie
PalmOS	KVM	CLDC
PalmOS	Midp	MIDP
WindowsCE 2.11	PersonalJava	PersonalJava
PalmOS, WindowsCE	IBM j9 VM	CLDC/MIDP
PalmOS	Jbed	CLDC/MIDP
PalmOS	Color KVM	CLDC, kAWT
Symbian EPOC	Symbian's Java	PersonalJava
WindowsCE 2.12 e 3.0	Jeole EVM	PersonalJava, EmbeddedJava
WindowsCE	crEme	
BlackBerry	J2ME	CLDC, MIDP
PalmOS, WindowsCE, PocketPC	Kada VM	PersonalJava, CLDC, CDC
WindowsCE	Chai VM	CLDC
SAVAJE	J2SE	Java 2
Zaurus	Zaurus	PersonalJava
PoketLinux	Kaffe	Java 2
Psion 5mx	JVM 1.00	
EPOC	Psion Java Platform	Java 1.1.4
WindowsCE	microJBlend	CLDC, MIDP

Volendo mantenere la compatibilità con le specifiche ufficiali di Java e confidando nello sviluppo della tecnologia, la scelta per lo sviluppo del prototipo del canale di comunicazione utilizzando il palmare ricade su un prodotto di SUN Microsystem.

Limitando la scelta alle VM compatibili con il Palm Vx (e cioè con PalmOS 3.0) si scopre che anche in casa SUN esiste più di un prodotto adatto ai nostri scopi.

Nella fattispecie troviamo due diverse VM:

- *KVM (Kilobytes Virtual Machine)*: è un porting della JVM progettato per girare su dispositivi dotati di memoria dell'ordine dei Kilobyte. Fu introdotta per la prima volta nel 1999 per dimostrare la possibilità di eseguire una VM ridotta su dispositivi tascabili.
- *MIDP Palm profile*: nato successivamente e destinato a rimpiazzare la prima KVM, rappresenta il tentativo di SUN di definire una piattaforma unica per lo sviluppo di applicazioni su dispositivi tascabili (compresi cellulari, palmari, agende, sim, ecc..)

Le differenze tra le due implementazioni sono evidenti e rivestono un ruolo fondamentale nella scelta della tecnologia da utilizzare per la realizzazione del canale mobile con palmare.

Il progetto KVM è stato realizzato basandosi sull'architettura AWT 1.0, cioè con una gestione centralizzata degli eventi.

Il modello di riferimento è quello delle Spotlet, ogni form ha il pieno controllo del display e si occupa dell'aggiornamento dei componenti posti su di essa.

Le uniche funzionalità di base implementate dalla VM sono:

- Primitive grafiche per disegnare sul display in coordinate assolute
- Gestione delle connessioni di rete e ad internet
- Lettura dello stato degli hotkeys
- Gestione degli eventi relativi alla penna (touch e move)

Questo modello presenta alcuni vantaggi:

- Estrema semplicità e compattezza della VM
- Semplicità del modello di gestione degli eventi
- Possibilità di espandere e, volendo, riscrivere completamente la libreria di oggetti grafici

La connettività viene garantita attraverso la libreria CLDC (Connected Limited Device Configuration) che fornisce un supporto di base ai protocolli socket ed HTTP. Per quanto riguarda MIDP, si tratta della nuova tecnologia di riferimento per J2ME. L'architettura di base si rivolge ai telefoni cellulari, fornendo un supporto di I/O tipico di questa tipologia di apparecchi:

Come già visto, La KVM (Kilobyte Virtual Machine) è stata presentata inizialmente nel 1999, ma fin dall'inizio ha avuto un carattere "dimostrativo", cioè voleva semplicemente mostrare la possibilità di realizzare una VM per dispositivi con poca memoria e ridotte capacità di connessione ed interazione.

Sulla base della KVM è stata successivamente sviluppato un framework detto KVM\_cldc (Connected, Limited Device Configuration), che costituisce una versione "lite" del CDC (Connected Device Configuration). Questa versione della virtual machine è nata proprio per supportare la connettività di dispositivi come i palmari.

Parallelamente (inizio anno 2000) il progetto Java One è stato commutato in MIDP (Mobile Information Device Profile), che aveva invece come target i telefoni cellulari di nuova generazione. Questo standard è stato recepito da produttori quali Nokia, Ericsson e Motorola.

Il problema è nato quando si è cercato di mettere assieme le due tecnologie in una logica unitaria. In effetti, mentre il progetto MIDP si consolidava ma restava vincolato alle ridottissime risorse dei cellulari, il progetto KVM\_CLDC riusciva a sfruttare in maniera piuttosto compiuta le potenzialità dei palmari, ma rimaneva carente dal punto di vista del networking (di fatto era supportato soltanto il servizio wireless del Palm VII). Dopo quasi un anno (ed innumerevoli segnalazioni da parte degli sviluppatori) SUN ha intrapreso una politica di graduale abbandono della tecnologia KVM\_CLDC in favore di MIDP. Sebbene non ci siano comunicati ufficiali in tal senso, appare evidente il fatto che i file del progetto KVM\_CLDC non sono più disponibili per il download (mentre tutta la documentazione è ancora disponibile).

Parallelamente il progetto MIDP si è trasformato, andando a costituire una piattaforma base per lo sviluppo di applicazioni per una vasta gamma di dispositivi, purtroppo le specifiche disponibili sono solo quelle di base, cioè quelle relative alle funzionalità minime comuni a tutti i dispositivi (cioè quelle dei cellulari), mentre la definizione di caratteristiche specifiche delle varie piattaforme verrà delegato alla stesura di particolari "configurazioni". Questo approccio non è stato ben accolto nella comunità JAVA; visto che i palmari hanno caratteristiche ben superiori agli altri dispositivi wireless questo ha di fatto ridotto tutta l'interfaccia utilizzabile a menu a scorrimento e pulsanti funzione.

L'ultimo aggiornamento risale al 2002, quando finalmente è stata rilasciata la configurazione MIDP per la linea Palm. Come prevedibile più che una implementazione completa e puntuale consiste nella realizzazione di una piattaforma Palm delle sole caratteristiche base del MIDP. Di fatto il progetto MIDP è ancora lontano dal completamento.

In teoria le piattaforme di riferimento della tecnologia Java dovrebbero essere 4:

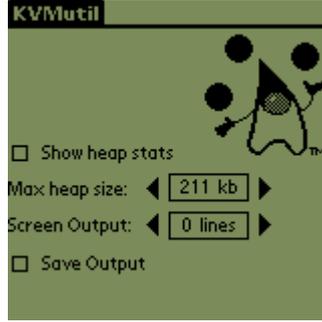
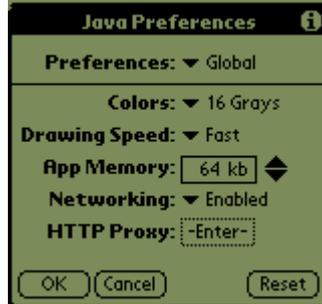
- 1) J2EE per le aziende e l'impresa
- 2) J2SE per l'uso personale
- 3) MIDP (KVM\_CLDC + MIDP configuration) per dispositivi wireless (cellulari Palmari ecc..)
- 4) JavaCARD per le smart card

Di fatto sussiste ancora una netta separazione tra i target di KVM\_CLDC e MIDP (che tuttavia include una versione di KVM)

Le principali differenze tra KVM e MIDP fino ad ora evidenziate possono essere riassunte come segue:

	KVM CLDC	MIDP
Interfaccia grafica	buona libreria di classi, anche estendibile, che ricalca del AWT 1.0	supporto decisamente scadente, i componenti non possono essere collocati liberamente nello spazio. Di fatto si lavora come su un cellulare
Input testuale	supporto di base per l'I/O testuale, senza la possibilità di utilizzare la tastiera software o di fare text editing avanzato	supporto alla tastiera software ed al text editing, ma solo su schermate dedicate esclusivamente all'input testuale, non è possibile introdurre un componente testuale all'interno di una pagina
Connessione alla rete	il collegamento viene stabilito all'apertura dell'applicazione, purché il processo di compilazione dichiari che il file PRC dovrà accedere alla rete	la connessione avviene all'occorrenza quando viene fatta richiesta l'apertura di un socket (o canale analogo)
Potocolli supportati	allo stato attuale sembra supportato solo il SOCKET	l'implementazione base richiede (ed implementa) solo il DATAGRAM, nessun supporto per un raw socket
Memoria disponibile	configurabile dalla VM, nell'ordine di qualche centinaio di Kb (600 - 700Kb nel caso del Palm Vx)	Al massimo 64k anche su un palm Vx con 8M
Tecnologie di riferimento	spotlet	midlet
Supporto allo sviluppo	solo shell di dos	semplice applicazione che consente la compilazione ed il deploy dei progetti. Ultimamente è stato integrato in Forte for Java (SUNOne ME)

Ciascuna VM deve essere installata sul Palm dato che i file PRC generati dalla compilazione richiedono le classi di libreria per funzionare correttamente. In entrambi i casi è disponibile un tool che permette di impostare alcune opzioni della VM tra cui la memoria utilizzabile dalle applicazioni.

		<p><b>KVM</b></p> <p>E' possibile impostare la massima memoria utilizzabile ed alcune opzioni di debug. La connessione alla rete è sempre disponibile e viene abilitata da un parametro durante la compilazione delle classi in PRC</p>
		<p><b>MIDP</b></p> <p>E' possibile selezionare la profondità cromatica, la memoria disponibile ed abilitare la connessione di rete</p>

Concludendo, la scelta ricade sulla “vecchia” KVM, che - anche se ormai poco supportata - presenta diversi e sostanziali vantaggi sulla concorrente MIDP:

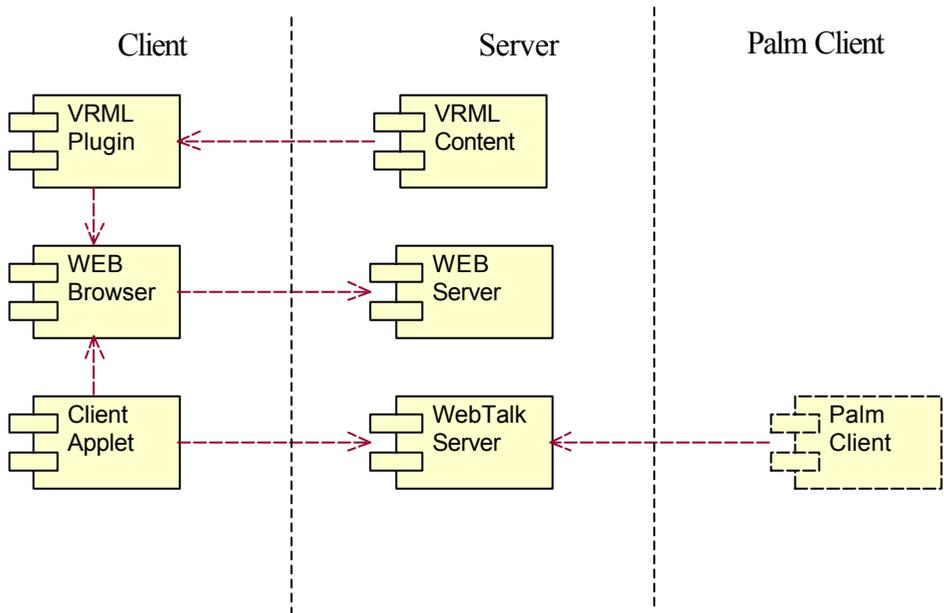
1. maggiore potenzialità grafiche
2. maggiore libertà nel posizionamento dei componenti e nella gestione dell'input
3. migliore sfruttamento delle risorse di memoria
4. maggiore velocità di esecuzione
5. modello ad eventi più versatile
6. supporto del raw socket, che è il mezzo più immediato per implementare un protocollo proprietario quale quello di WebTalk

## 7.4 Implementazione del canale per Palmare

### 7.4.1 Architettura fisica

Per l'implementazione a livello fisico del canale radiomobile basato su palmare Palm, si è scelto di realizzare il server attraverso l'architettura di stato condiviso realizzata in formato "leggero" e destinata al colloquio con applet Java di controllo dei plug-in di visualizzazione VRML, configurazione architetturale denominata WebTalk-I.

Poiché infatti la piattaforma palmare ha la possibilità di collegarsi direttamente al gestore dello stato condiviso (al contrario dei terminali WAP), tuttavia non si è ritenuto opportuno caricare il palmare di processi computazionalmente pesanti per disincapsulare entità passate come oggetti serializzati contenenti quaternioni, ma elaborare PDU molto più leggere, come quelle utilizzate nell'architettura WebTalk-I.



**Fig. 7-24: lo schema dei componenti di riferimento per il canale PalmOS. Il client su palmare attiva una connessione diretta con il gestore di stato condiviso.**

Occorre poi considerare che, diversamente da quanto accade per il canale con rappresentazione 3D, non è possibile integrare il client di canale con un browser internet, e questo significa che non è possibile rappresentare direttamente il contenuto tridimensionale. Per questo motivo il canale palmare non accede via HTTP ai contenuti geometrici serviti attraverso la rete, poiché non sarebbe comunque in grado di rappresentarli nella forma tridimensionale.

Una conseguenza del fatto che il client palmare attiva una connessione diretta al server, è che occorre una fase di configurazione che specifichi i seguenti dati:

- l'indirizzo del server webtalk
- la porta su cui ascolta il server
- il mondo a cui si vuole accedere
- l'indirizzo della geometria VRML dell'avatar da cui si vuole essere rappresentati nel mondo virtuale

E' pertanto necessario prevedere una prima fase all'inizio dell'esecuzione del client in cui l'utente le specifichi esplicitamente. Per evitare l'immissione ad ogni sessione dei presenti dati, e creare una identificazione fissa tra un particolare palmare utilizzato per la connessione e la rappresentazione all'interno dello spazio condiviso, viene utilizzato il sistema PalmOS per la memorizzazione di dati permanenti, cioè dati che sopravvivono agli spegnimenti del palmare.

La tecnologia PalmDB permette ad ogni applicazione di creare e mantenere aggiornato un database di dati non tipizzati.

Ogni database contiene una lista ordinata di oggetti che possono essere serializzati; in questo modo è possibile definire una classe che incapsuli il tipo di dato che si desidera salvare ed inserirne le istanze all'interno del PalmDB.

L'utente può quindi mantenere un database dei server, aggiornarlo ed utilizzarlo per connettersi rapidamente ai server preferiti.



**Fig. 7-25: i dati per la connessione diretta al gestore dello stato condiviso sono selezionabili da parte dell'utente all'interno del client realizzato per la connessione**

Le modalità di collegamento del palmare alla rete di distribuzione ed il successivo raggiungimento del server sono sostanzialmente due:

- attraverso un accesso radiomobile, utilizzando un modem GSM o GPRS (e in un vicino futuro UMTS). Nella soluzione GSM, la connessione si è rivelata eccessivamente lenta, e soprattutto costosa. Questo problema si è nettamente attenuato con la connessione utilizzando un modem GPRS, anche se la risposta

del sistema rimane non ottimale, e il problema del costo rimane un parametro sensibile.

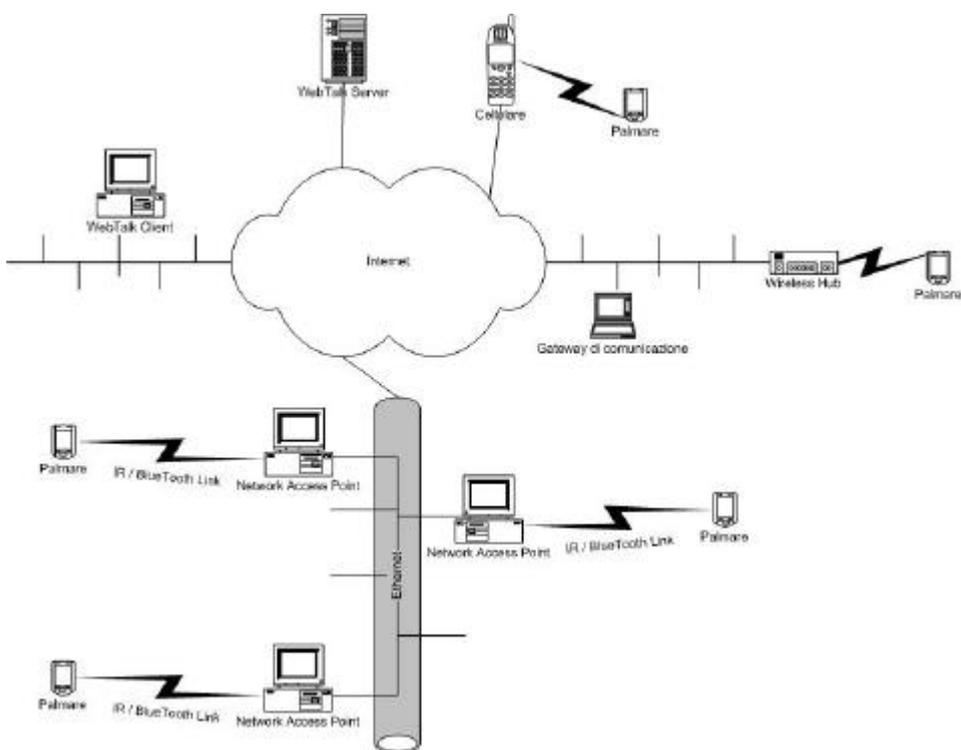
- utilizzare un computer collegato alla rete come gateway per la connessione del palmare e sfruttare la connessione seriale del cradle per emulare una connessione wireless. Questa soluzione consente comunque una mobilità anche se di raggio limitato, utilizzando uno short-link radio tra il palmare stesso ed il gateway di comunicazione. Una soluzione tecnologica di crescente diffusione è l'utilizzo di wireless LAN e della corrispondente scheda wireless disponibile per alcuni modelli di palmari. In questo modo il palmare diventa un componente mobile di una rete fissa con accesso a larga banda sulla rete.
- Una terza soluzione, che consente di superare il problema del raggio limitato ma è di implementazione locale, è di disseminare l'ambiente geografico di collaborazione mobile di NAP (Network Access Point) dotati di un interfaccia radio a corto raggio (ad esempio un collegamento IR, o Bluetooth). Gli utenti che si muovono all'interno dell'area geografica possono entrare in rete puntando il proprio palmare verso l'access point (occorre una visibilità precisa con un link IR, e meno precisione con un link Bluetooth).

Le operazioni di comunicazione (ricezione e trasmissione) tra il palmare e il gestore dello stato condiviso debbono procedere parallelamente ed in maniera asincrona, e prevedere momenti in cui la connessione risulta instabile (ad esempio nei momenti di switch tra un access point e l'altro, oppure tra una cella GPRS e l'altra in caso di ampia mobilità geografica). Queste considerazioni, portano a strutturare il software client su due thread che girino contemporaneamente sul palmare: il primo si occupa di gestire l'acquisizione ed il trattamento dell'input dall'utente mentre l'altro si occupa di gestire i messaggi che provengono dal server. In questo modo eventuali delay in ricezione o trasmissione non bloccano completamente l'esecuzione del sistema.

#### **7.4.2 Strato di Presentazione**

La rappresentazione dello stato condiviso sul canale palmare può essere organizzata in modo più complesso rispetto al canale WAP, grazie a maggiori possibilità di visualizzazione, una maggiore dimensione del display, e non ultimo la possibilità di collegarsi direttamente allo stato condiviso.

All'utente può essere offerta un'ampia finestra di chat, con funzionalità di scrolling e scrittura dei messaggi mediante lo stilo. Inoltre viene fornita una proiezione bidimensionale dello spazio condiviso geometrico, con la mappatura delle posizioni propria e degli altri utenti, con le modalità descritte in seguito. Viene poi fornita una lista di oggetti condivisi, con cui è possibile operare alcune interazioni di base. Lo spazio disponibile per questa rappresentazione deve essere gestito in un'area di 160 x 160 pixel.



**Fig. 7-26: le possibili connessioni di rete per un canale basato su palmare sono attraverso un link radiomobile, oppure attraverso un gateway agganciato alla rete fissa (eventualmente con un radio short-link realizzabile con una wireless LAN, oppure disseminando sull'area geografica di esplorazione una serie di access point dotati di connettività a corto raggio).**

Per quanto riguarda lo scambio di messaggi, è possibile sfruttare circa la metà dello spazio disponibile per visualizzare un certo numero di righe di testo. Inoltre è possibile inserire un componente di testo in cui l'utente può inserire le frasi che intende usare in chat ed un pulsante che ne consenta l'invio al sistema.

La visualizzazione dello spazio geometrico condiviso deve adottare una proiezione bidimensionale. Due sono le possibili scelte da considerare:

- *una visualizzazione in prima persona* in cui proiettare gli avatar in maniera piuttosto simile a quella utilizzata dal visualizzatore VRML (prospettica), in questo caso vengono visualizzati solo gli oggetti di fronte all'utente e le due coordinate fisiche rappresentano l'altitudine e lo spostamento relativi rispetto all'avatar.

- *una proiezione ortogonale* in cui utilizzare una visuale “dall’alto” della scena, come se si osservasse il mondo stando al di sopra dell’avatar e si guardasse verso il basso, con l’avatar in centro allo schermo.

Appare evidente come una proiezione ortogonale sia più leggera dal punto di vista computazionale rispetto ad una proiezione prospettica, infatti mentre una proiezione ortogonale richiede uno scalamento (per il fattore di vista) ed al più una rotazione, la proiezione prospettica aggiunge una serie di divisioni e prodotti per eseguire le proporzioni al piano di proiezione.

In secondo luogo una proiezione prospettica dimezzerebbe lo spazio visualizzabile, infatti verrebbe ignorato tutto il semispazio dietro l’avatar e parte dello spazio di fronte (quello al di fuori del frustum visivo) introducendo inoltre ulteriori appesantimenti dovuti alla necessità di un algoritmo di clipping. Questi inconvenienti assumono un peso ancora maggiore se si valutano attentamente i limiti del Palm, tra cui spicca l’assenza di una matematica in virgola mobile, e la particolare lentezza dell’operazione di divisione, che conviene pre-elaborare sostituendo ciascun fattore di divisione con un corrispondente fattore moltiplicativo inverso.

Scegliendo di adottare una visualizzazione dall’alto, il client del canale palmare deve eseguire una mappatura tra il volume di visualizzazione tridimensionale considerato ed il piano di clipping bidimensionale sul display del palmare.

La finestra di proiezione sul client viene definita mediante le sue dimensioni di altezza e larghezza, ed il relativo aspect ratio, che determina il fattore di schiacciamento della proiezione, dato dal rapporto tra la sua larghezza e la sua altezza. La reale dimensione del sottospazio da proiettare viene poi definita da un fattore di scala detto *scope* che indica la profondità (in unità del mondo virtuale<sup>30</sup>) del campo visuale; in sostanza il volume di renderizzazione risulta centrato sull’avatar dell’utente, profondo  $2 * scope$  unità, e convenzionalmente alto quanto profondo, e largo in modo da rispettare l’aspect ratio del componente di visualizzazione.

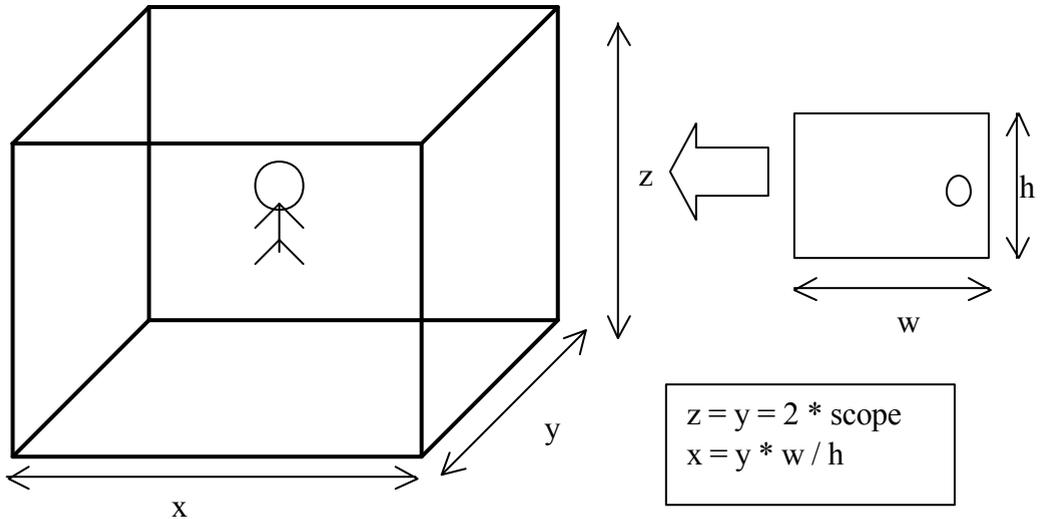
Resta da scegliere la profondità di vista (*scope*).

Non è possibile scegliere una profondità che vada bene per tutte le utenze e, soprattutto, per tutti i mondi, dato che il significato dato alla unità di rappresentazione può variare rispetto alle dimensioni dell’avatar.

All’interno dello stesso mondo potrebbe poi essere utile sfruttare viste a zoom variabile; si immagini ad esempio di essere in mezzo ad una piazza di grandi dimensioni, per avere un’idea di come si sono disposti gli utenti ed identificare pertanto i gruppi di utenti è utile visualizzare tutta la piazza in una sola schermata ed è quindi necessario avere uno *scope* molto alto (o un livello di zoom molto basso se si preferisce), viceversa una volta avvicinarsi al gruppo è indispensabile poter vedere i dettagli delle posizioni di ciascun avatar per cercare altri utenti o semplicemente seguire più in dettaglio le interazioni tra gli altri utenti.

---

<sup>30</sup> *Le rappresentazioni tridimensionali, e le relative API di modellazione e rendering, non adottano mai una unità di misura definita (quale metro o centimetro), ma rappresentano le geometrie in termine di una generica “unità”, cui il modellatore può assegnare il rapporto di scala desiderato nella rappresentazione.*



**Fig. 7-27: proiezione sul piano di clipping bidimensionale della scena, necessaria alla rappresentazione modificata per il canale palmare. Il volume di visualizzazione per la mappatura si ottiene dai parametri di aspect (fissati sulle dimensioni del palmare) e sul parametro di scope (legato al fattore di zoom).**

Occorre inoltre rendere la velocità di spostamento degli avatar proporzionale allo scope, e si può sfruttare la dinamicità dello zoom per agevolare grandi spostamenti quando necessario, o conversamente permettere posizionamenti fini dove richiesto. Per questi motivi è opportuno inizializzare lo scope ad un valore plausibile (dell'ordine di qualche decina di unità) per permettere subito all'utente di avere un'idea della disposizione degli avatar, ma dare la possibilità all'utente stesso di aumentare o diminuire lo zoom a seconda delle esigenze.

A tal fine è stato aggiunto un controllo di zoom in ed uno di zoom out sul componente di visualizzazione; sperimentalmente una buona funzione di zoom appare essere la legge esponenziale in quanto cresce in maniera sufficientemente rapida da permettere cambi di zoom di fattori elevati con pochi comandi. Applicando allo scope la seguente funzione:

$$f(x) = 2^x$$

lo scope raddoppia ad ogni operazione di zoom out e si dimezza (fino ad un minimo di 1 unità) ad ogni operazione di zoom in.

Resta da rappresentare la terza dimensione, cioè l'altezza relativa degli avatar rispetto all'osservatore. E' semplice rappresentare questa terza dimensione dividendo il volume

di renderizzazione in 3 livelli paralleli, uno in cui si trova l'osservatore, uno sopra l'osservatore ed uno sotto di esso.

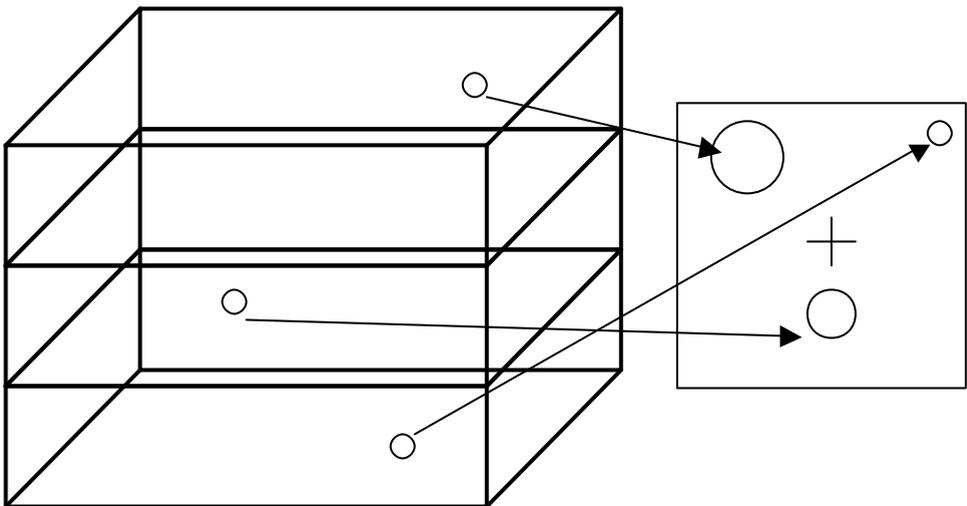
Una volta fatto ciò basta associare tre icone differenti (o semplicemente tre dimensioni differenti alla stessa icona) per i tre sottospazi: così facendo gli utenti che si trovano nel terzo superiore del volume di renderizzazione verranno visualizzati più grandi di quelli che si trovano allo stesso livello dell'osservatore che a loro volta saranno più grandi di quelli che stanno più in basso. Con questo accorgimento è facile rappresentare le quote degli avatar, anche se in maniera estremamente rudimentale.

Data la limitatezza del display si è scelto di visualizzare gli utenti come semplici rettangoli, senza indicazioni sul loro orientamento.

E' inoltre necessario poter associare ogni rettangolo con l'utente corrispondente, in caso contrario la visualizzazione del mondo 3D non darebbe informazioni circa la posizione di utenti specifici e sarebbe equivalente ad una cartina muta.

Per fare questo è sufficiente scrivere sopra all'utente desiderato il nome con cui è conosciuto al sistema, scrivere il nome sopra ogni utente occuperebbe troppo spazio e riempirebbe la mappa di scritte rendendola illeggibile; è più conveniente permettere all'utente di selezionare un solo avatar per volta e visualizzare il nome solo sopra l'avatar selezionato. Così facendo si permette di:

- seguire agevolmente l'interlocutore attraverso i suoi spostamenti
- identificare a turno tutti gli utenti
- mantenere ordinata e pulita la visualizzazione del mondo



**Fig 7-28: La rappresentazione della quota degli utenti nello spazio condiviso adottata dal canale Palm.**

### 7.4.3 Strato di Navigazione

Le funzioni di navigazione previste per il canale Palm non prevedono l'uscita dalla visualizzazione principale, comprendente l'area di conversazione, la mappa bidimensionale e l'elenco degli oggetti.

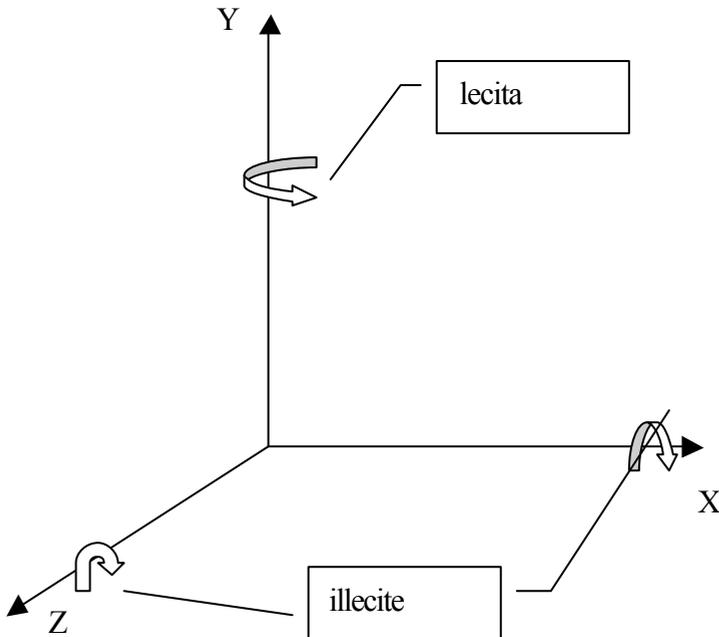
Non vengono forniti pertanto sistemi per cambiare di schermata, ma tutti i comandi di navigazione servono per attivare una delle tre aree (messaggi, mappa, oggetti), ed eventualmente per comandare uno spostamento del proprio avatar nello spazio o per interagire con un oggetto.

Per quanto riguarda gli spostamenti dell'utente del palmare, occorre sfruttare tutti i controlli che il dispositivo mette a disposizione in modo da non aggiungere ulteriori pulsanti sul display. E' indispensabile permettere all'utente di posizionare l'avatar in qualunque punto dello spazio senza però complicare eccessivamente l'interfaccia e, soprattutto, evitando che una successione di rotazioni faccia perdere all'utente l'orientamento. Quest'ultimo è un problema che si riscontra anche nei navigatori più complessi come i browser VRML, nel caso di un palmare questo avrebbe però un esito particolarmente negativo sulla navigabilità del mondo.

La perdita di orientamento da parte dell'utente è facilmente risolvibile nel caso di un'interfaccia grafica ricca e completa che fornisce indicazioni visive precise della situazione, ma nel caso di una semplice visualizzazione 2D su di un palmare questo non avviene ed il disorientamento comporterebbe una serie di comandi di navigazione errati che porterebbero ben presto alla saturazione delle già scarse potenzialità di calcolo del dispositivo e farebbero con ogni probabilità desistere l'utente dall'utilizzarlo.

Per ovviare a questi problemi è possibile introdurre dei vincoli alle configurazioni possibili in cui si può trovare l'avatar di un utente Palm.

In particolare è opportuno vietare ogni rotazione lungo gli assi X e Z, questo significa che l'utente mantiene sempre lo sguardo verso l'orizzonte e può semplicemente ruotare lo sguardo a sinistra e a destra.



**Fig. 7-29: restrizione navigazionale imposta dalle caratteristiche del canale Palm**

Una soluzione ancora più restrittiva sarebbe quella di consentire solo traslazioni e vietare anche le rotazioni lungo l'asse Y, questa scelta renderebbe però eccessivamente rigido il comportamento degli avatar dei palmari : ad esempio non sarebbe possibile per un utente palmare mettersi “faccia a faccia” con un interlocutore.

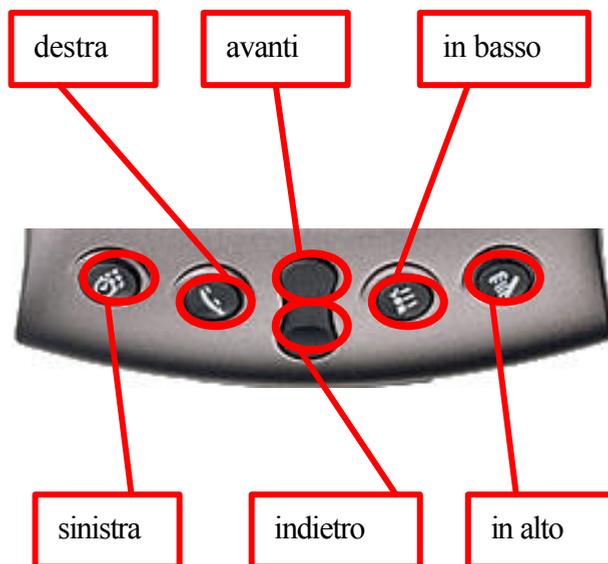
Mantenere la rotazione lungo l'asse Y ed utilizzare delle traslazioni rigide lungo tale asse per prendere o perdere quota appare pertanto il miglior compromesso tra leggerezza e solidità dell'interfaccia e libertà di movimento.

Sono pertanto 6 i movimenti consentiti ad un utente palmare:

1. avanzare: interessa il solo piano X/Z a seconda dell'angolo attuale rispetto ad Y
2. indietreggiare: interessa ancora il solo piano X/Z
3. ruotare a sinistra: varia il solo angolo libero dell'avatar
4. ruotare a destra: varia il solo angolo libero dell'avatar
5. alzarsi: aumenta la quota mediante una traslazione rigida lungo l'asse Y
6. abbassarsi: riduce l quota mediante una traslazione rigida lungo l'asse Y

Per permettere all'utente di impartire questi comandi si utilizzano le hot key, cioè i tasti funzione che si trovano nella parte inferiore del palm, al di sotto dello schermo e della zona di scrittura con lo stilo.

I tasti sono appunto 6, posizionati in maniera particolarmente adatta per l'assegnazione delle funzioni viste. Le funzioni di rotazione sono assegnate ai due tasti di sinistra, i tasti verticali in centro controllano l'avanzamento e l'arretramento dell'avatar, mentre i due tasti di destra permettono di modificare la quota.



**Fig 7-30: per semplificare l'interfaccia di navigazione, si assegnano funzioni predefinite ai tasti messi a disposizione dalla periferica usata nel canale palmare.**

Tutti gli spostamenti sono quantizzati, cioè si fissa un passo sia nelle rotazioni che nelle traslazioni, in questo modo si riduce il numero di messaggi generati e si rende più immediato l'utilizzo da parte dell'utente.

I passi di discretizzazione sono fissi per quanto riguarda le rotazioni e dinamici per quanto riguarda le traslazioni.

Nel caso di traslazioni lo spostamento è proporzionale allo scope attuale, infatti non avrebbe senso fare eseguire piccoli spostamenti di pochi centimetri quando lo spostamento minimo apprezzabile è nell'ordine delle decine di metri e, viceversa, permettere grandi spostamenti quando lo zoom è particolarmente alto comporterebbe inevitabilmente lo smarrimento dell'utente che si vedrebbe istantaneamente catapultato in zone del mondo non pertinenti alla sua scala di osservazione.

Tipo di movimento	Passo di aggiornamento
Avanti	+ Scope / 4
Indietro	- Scope / 4
Rotazione a destra	+ 45 °
Rotazione a sinistra	- 45 °
In alto	+ Scope / 4
In basso	- Scope / 4

La navigazione dell'utente si basa anche sulle posizioni occupate puntualmente dagli altri utenti nello spazio. L'aggiornamento visivo dello stato condiviso deve essere implementato tenendo presente le limitate possibilità computazionali della periferica palmare. In particolare le operazioni di lettura dal socket di connessione risultano all'interno della Virtual Machine sul palmare particolarmente lente, pertanto occorre particolare cura nell'ottimizzare la trasmissione e la ricezione necessaria all'aggiornamento.

Poiché la maggioranza dei messaggi scambiati con il server riguarda lo spostamento degli avatar, appare chiaro che sia questo tipo di messaggi l'obiettivo dell'operazione di filtraggio. Occorre dunque aumentare la granularità degli spostamenti per ignorare spostamenti piccoli o comunque insignificanti al fine della visualizzazione sul palmare.

Questo può essere ottenuto introducendo una segmentazione dello spazio virtuale in blocchi di dimensione finita. Verranno inviati al palmare solo gli spostamenti che comportano un cambio di segmento.

In questo modo si permette ad ogni utente di muoversi liberamente all'interno di piccoli cubi senza che i palmari connessi al sistema se ne accorgano.

Si noti che la segmentazione dello spazio non equivale ad imporre uno spostamento minimo, in questo caso, infatti, uno spostamento anche molto consistente ma composto da molti piccoli passi non verrebbe rilevato comportando una grossa incongruenza tra i mondi percepiti dai palmari e la reale posizione degli avatar.

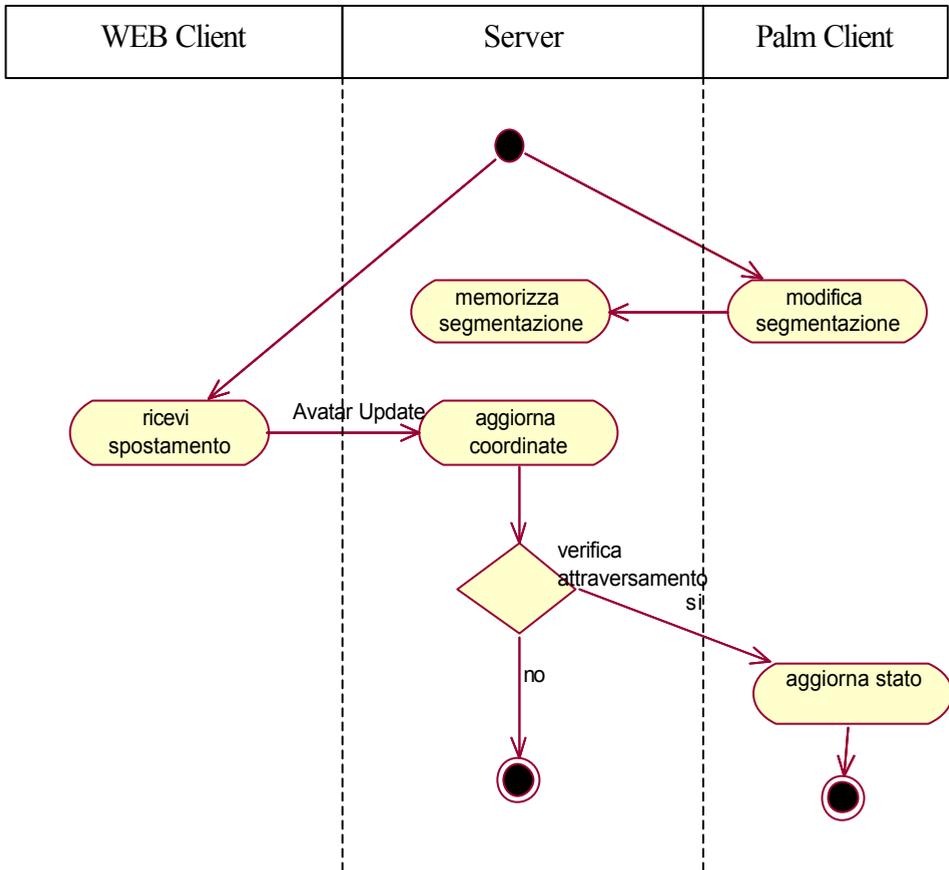
La granularità della segmentazione si comporta come il passo degli spostamenti, è cioè dinamica e definita come un sottomultiplo dello scope.

Questo risponde al fatto che più è alto lo scope (e cioè basso lo zoom) più è possibile ignorare gli spostamenti, mentre quando si sta guardando la disposizione degli avatar nel dettaglio anche il minimo spostamento può comportare una variazione qualitativa della scena.

La segmentazione dello spazio virtuale può essere eseguita in due sistemi di coordinate differenti:

- *il sistema di coordinate assoluto* del mondo: questo significa che i cubi in cui viene suddiviso lo spazio partono dall'origine degli assi e si espandono lungo gli assi X, Y e Z del mondo.

-



**Fig. 7-31: diagramma delle attività necessarie a gestire il movimento per segmentazione. La granularità di segmentazione è selezionabile lato client.**

- *il sistema relativo all'osservatore:* in questo caso la segmentazione parte dalla posizione dell'utente del palmare e si estende lungo le tre dimensioni come le percepisce l'utente, cioè verso destra, l'alto e la profondità

Il caso del sistema di riferimento assoluto è sicuramente il più semplice da gestire: ogni volta che il server riceve uno spostamento incomincia l'operazione di dispatch, e nell'inoltro a una connessione palmare si verifica se lo spostamento varca un segmento e solo in questo caso lo si invia alla connessione in esame.

Il principale limite del sistema di coordinate assoluto è che, non essendo sensibile alla reale posizione dell'avatar del palmare, rischia di trascurare spostamenti che, nonostante siano modesti per entità, hanno un grosso peso qualitativo sulla scena e vengono percepiti come informazioni importanti dal navigatore.

Un caso critico è quello in cui l'avatar passa da davanti a dietro il palmare oppure dalla sua sinistra alla sua destra.

Gli utenti sono particolarmente sensibili al quadrante relativo in cui si trovano gli avatar. Ad esempio il fatto che l'avatar "Leonardo" si trovi immediatamente di fronte e a destra rispetto all'osservatore è molto più significativo del fatto che disti dall'osservatore 5.4 metri.

Utilizzando un sistema di coordinate assoluto per la segmentazione, ci si può trovare nel caso in cui l'utente del palmare si trovi all'incirca al centro di un segmento, in tal caso un eventuale altro utente può spostarsi all'interno di questo segmento generando modifiche anche importanti della scena senza che il palmare ne venga avvisato.

Al contrario, utilizzando un sistema di riferimento centrato sull'osservatore si rilevano tutti i cambi di quadrante rispetto all'utente e non si perdono messaggi qualitativamente importanti.

Nell'esempio di segmentazione riportato in figura si evidenzia come nel caso di coordinate assolute (sinistra) l'utente possa girare liberamente intorno all'osservatore senza generare cross dei segmenti, mentre utilizzando un sistema relativo (già ruotato nella direzione di osservazione) lo stesso movimento genera ben tre aggiornamenti.

Il sistema di segmentazione secondo il riferimento relativo non è però esente da controindicazioni e problemi:

- Richiede maggiori calcoli, nella fattispecie richiede una roto/traslazione per ogni connessione palmare e questo può impattare negativamente sulle performance del server.
- Esistono delle corse critiche in cui gli spostamenti sono tali da generare errori

Le corse critiche (o casi critici) nascono dal fatto che la segmentazione segue l'osservatore nei suoi spostamenti.

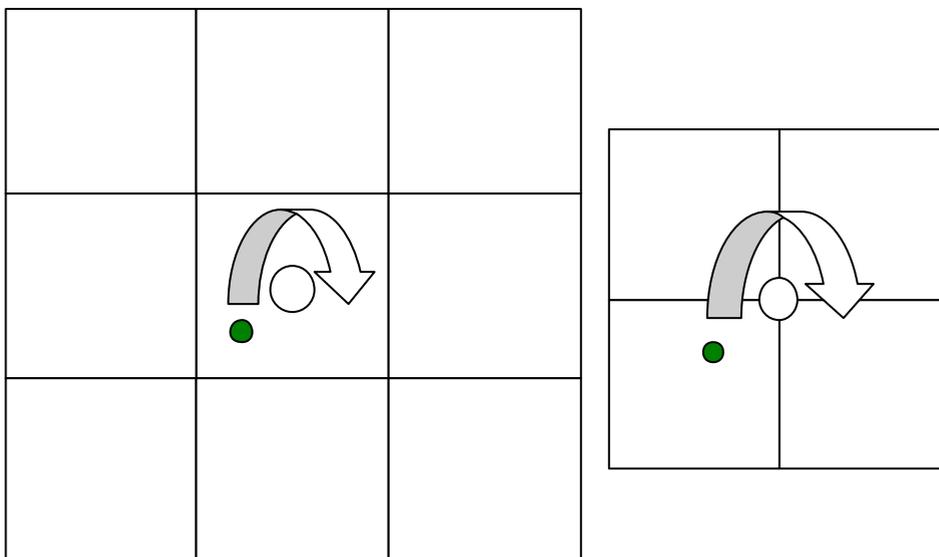
*Esempio:* Si immagini di avere impostato blocchi di lato pari a 5 unità e si riduca il problema ad un caso monodimensionale, per ovvi motivi di semplicità.

L'utente del palmare sia inizialmente nell'origine ed un altro utente sia di fronte a lui, diciamo a 2 unità di distanza.

A questo punto l'utente del PC avanza di 2 unità portandosi a 4 unità dall'osservatore. Questo spostamento non fa superare la distanza limite di 5 unità e non è pertanto inviata.

Ora il palmare avanza di 2 unità neutralizzando l'allontanamento del compagno di navigazione e ripristinando le distanze reciproche.

Di nuovo l'utente del PC si allontana di 2 unità e di nuovo il palmare avanza di altrettanto.

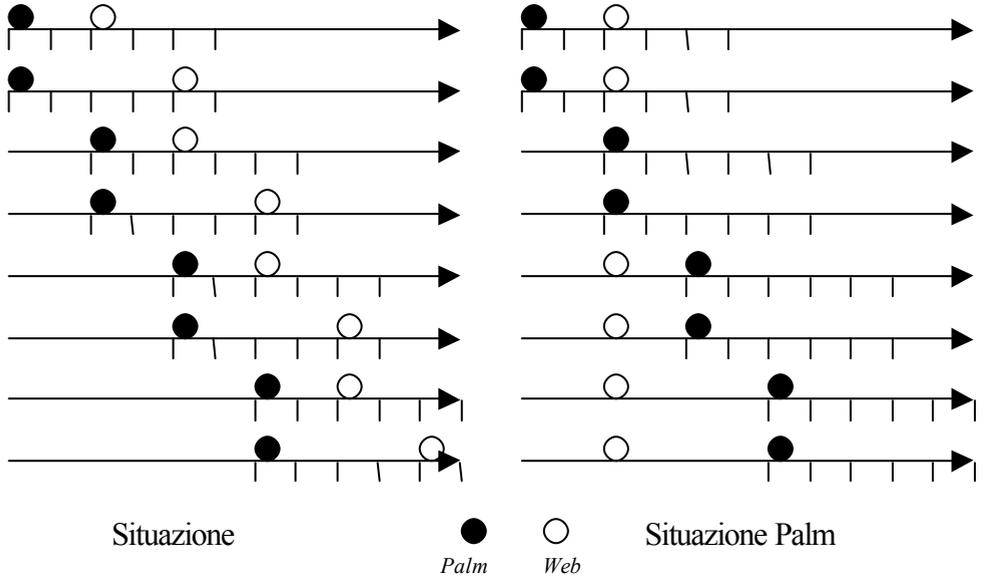


**Fig 7-32: confronto tra il numero di attraversamento di segmenti in un sistema di segmentazione a coordinate assolute (sinistra) ed uno a coordinate relative alla posizione corrente dell'utente (destra). Usando un sistema a coordinate relative (destra), risulta più precisa la rilevazione di eventi sensibili in quanto prossimi all'osservatore.**

Questa catena di spostamenti può andare avanti all'infinito senza che il palmare ne venga informato, dato che ciascuno spostamento non varca i limiti del segmento. Questo significa che, per esempio dopo 10 spostamenti, il palmare avrà ignorato gli spostamenti dell'utente PC per un totale di ben 20 unità e lo visualizzerà dietro di sé, con un errore decisamente intollerabile anche da un punto di vista qualitativo.

Per risolvere questo grave inconveniente sarebbe necessario:

- valutare gli attraversamenti non rispetto all'ultimo aggiornamento ricevuto dal server, ma dall'ultimo aggiornamento effettivamente inviato alla specifica connessione palmare, ma questo significherebbe mantenere una copia dello stato del mondo per ciascun palmare connesso e questo farebbe crescere come  $O(n^2)$  l'occupazione di memoria rispetto al numero di connessioni.
- richiedere, secondo opportune politiche, l'aggiornamento delle posizioni degli avatar anche quando a muoversi è l'utente su palmare; ovviamente questo genererebbe un tale traffico aggiuntivo da vanificare il vantaggio del filtraggio dei messaggi.



**Fig 7-33: una condizione di corsa critica verificabile utilizzando unicamente un sistema di segmentazione relativo alla posizione dell'utente del canale Palm.**

Per ridurre gli svantaggi dei due approcci conviene dunque utilizzare contemporaneamente entrambi i sistemi di riferimento, ma limitandosi ad un uso localizzato del sistema di coordinate relative.

Si può cioè utilizzare globalmente una segmentazione nello spazio assoluto, limitando così l'occupazione di memoria e risolvendo i casi patologici di cui sopra, ma affiancare localmente, nella posizione corrente dell'utente, un'analisi nello spazio relativo per forzare gli aggiornamenti nei casi in cui lo spostamento faccia cambiare quadrante relativo all'avatar. Si utilizza quindi lo spazio relativo solo per rilevare i cambiamenti qualitativi nella posizione relativa.

Così facendo si riesce a mantenere tutti i vantaggi dell'approccio assoluto, assicurando comunque il corretto posizionamento nei quadranti anche per piccoli spostamenti.

L'aggiornamento viene dunque inviato o quando si cambia segmento assoluto oppure quando si passa da di fronte a dietro l'osservatore, o da sinistra a destra o da sopra a sotto. La segmentazione così ottenuta genera, nella maggioranza dei casi, lo stesso numero di messaggi che genererebbe una impostazione a coordinate assolute.

Un altro aspetto di ottimizzazione riguarda la gestione delle rotazioni pure durante il movimento.

I visualizzatori tridimensionali gestiscono le rotazioni in maniera estremamente puntuale, generando un grande numero di messaggi di aggiornamento per consentire una corretta interpolazione dei movimenti.

Dalle esperienze di navigazione risulta che gran parte dei messaggi generati dalla navigazione tridimensionale si riferiscono a rotazioni pure, cioè rotazioni sul posto degli avatar; che si verificano, ad esempio, quando un utente esamina un oggetto, cerca un compagno o semplicemente si guarda attorno.

I messaggi di rotazione sono contemporaneamente i più numerosi (e pertanto dannosi ai fini delle performance del palmare) ed i più inutili.

Poiché però, a causa della risoluzione del dispositivo, gli utenti vengono visualizzati come semplici punti o rettangoli ed il loro orientamento non ha alcun impatto sulla visualizzazione del Palm, questi messaggi dovrebbero essere ignorati. Il meccanismo della segmentazione del mondo elimina automaticamente tutti questi messaggi: una rotazione pura infatti non genera spostamenti e lascia sempre l'avatar nello stesso segmento.

Grazie all'introduzione della segmentazione è stato possibile ridurre il numero di messaggi di un fattore variabile tra 6 e 10, rendendo possibile una navigazione agevole anche mediante palmare.

#### **7.4.4 Strato funzionale**

Una delle funzionalità offerte in più dal canale palmare rispetto al canale WAP è la possibilità di interazione con gli oggetti. Il tipo di interazione offerta è piuttosto basilare, data l'impossibilità di visualizzare in modo dettagliato le geometrie degli oggetti.

In particolare si è ridotta l'interazione ai soli eventi di click, cioè all'attivazione di sensori di pressione; a questi possono poi essere collegate animazioni ed azioni del tipo più disparato.

Una soluzione soddisfacente consiste nel visualizzare la lista completa degli oggetti condivisi presenti nel mondo e permettere all'utente di attivare ogni oggetto semplicemente cliccandoci sopra con la penna del palmare. Il supporto di input del palmare non permette di rilevare il rilascio della penna, per questo motivo si è preferito associare al click su un oggetto della lista l'attivazione dell'oggetto relativo per una durata standard di un secondo.

Viceversa, quando un utente attiva un oggetto condiviso mediante un click, ad esempio con il puntatore del mouse, l'utente del palmare può vedere selezionarsi nella lista il nome dell'oggetto in risposta all'attivazione da parte dell'utente. In questo caso la sincronizzazione è più fine nel senso che l'utente del palmare può vedere selezionato l'oggetto finché l'utente che ha generato l'evento non rilascia l'oggetto stesso.

Questa modalità di cooperazione è estremamente semplificata rispetto a quelle possibili in un normale mondo nel canale con rappresentazione 3D, ma è comunque un buon punto di incontro tra la capacità espressiva del sistema e le limitazioni dello strumento: tuttavia il principale limite funzionale del client risiede nell'insieme di informazioni che possono essere rese in maniera grafica all'interno del componente di visualizzazione.

Utilizzando la proiezione ortogonale è possibile:

- visualizzare un'area del mondo condiviso che circonda l'utente a 360° nelle tre dimensioni
- rendere con precisione la disposizione degli utenti nel piano X/Z
- rendere in maniera approssimata ma immediata ed intuitiva la disposizione lungo l'asse Y
- selezionare un utente visualizzandone il nome per seguirlo negli spostamenti e distinguerlo dagli altri utenti del sistema

Le uniche entità visualizzate sono gli utenti del sistema, manca invece completamente la rappresentazione degli oggetti sia condivisi che non.

Questo significa che non è possibile informare l'utente del palmare su:

- posizione degli oggetti condivisi
- esistenza e posizione di oggetti non condivisi
- esistenza e posizione di muri o barriere all'interno del mondo

Tutto ciò porta ad una generale riduzione del dominio funzionale degli avatar su palmare che, non potendo ottenere informazioni circa le caratteristiche fisiche e geometriche del mondo virtuale, possono eseguire operazioni e spostamenti incompatibili con i vincoli a cui invece sono assoggettati gli utenti web.

#### **7.4.5 Strato collaborativo**

Le funzionalità collaborative previste per il canale palmare coincidono sostanzialmente con quelle fornite dal gestore dello stato condiviso semplificato utilizzato per la navigazione diretta attraverso applet (architettura WebTalk-I). Pertanto è possibile navigare e colloquiare mediante uso di gruppi, formare e distruggere gruppi, spostarsi da un gruppo ad un altro. Per quanto riguarda la funzionalità di gruppo guidato, è possibile agganciarsi ad un altro utente, nel qual caso i movimenti sul palmare verranno aggiornati seguendo quelli dell'utente cui ci si è agganciati.

Occorre tuttavia osservare come la mancanza di effettive informazioni geometriche all'interno dello spazio condiviso rende la collaborazione in termini spaziali piuttosto esigua e priva del significato effettivo che assume all'interno dell'ambiente tridimensionale completo.

### 7.4.6 Esempio applicativo

Nel seguito si descrive una breve navigazione effettuata utilizzando due canali: il canale palmare descritto nel presente capitolo, ed un canale a navigazione tridimensionale con plug-in VRML.

Dal palmare PalmOS è possibile lanciare l'applicazione scegliendone la relativa icona.

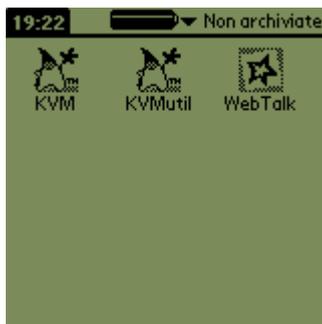


Fig 7-34: Picona di avvio del canale di comunicazione su Palm OS

Nella schermata principale, che si apre dopo l'accettazione della connessione da parte del server, vengono offerti tutti i controlli di cui dispone un utente i palmare:

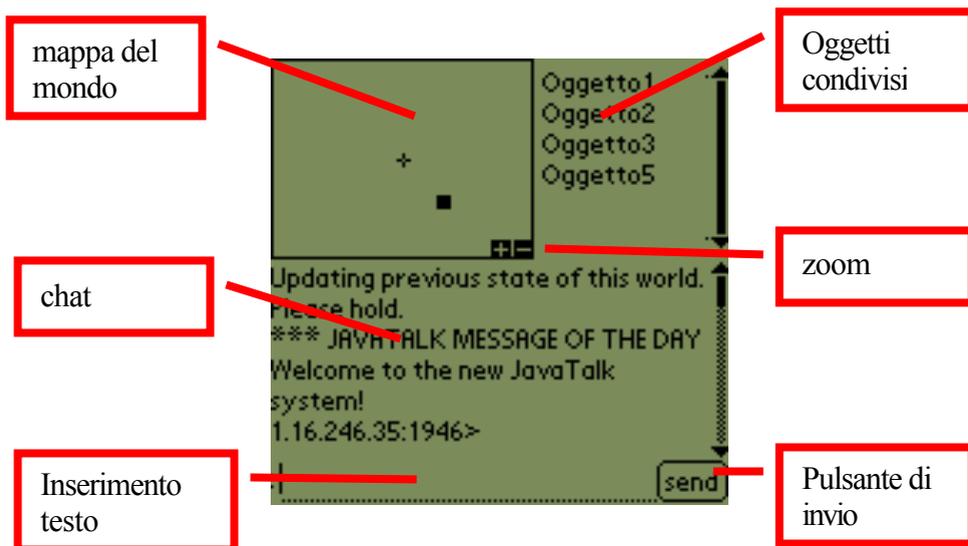


Fig 7-35: Lo spazio condiviso rappresentato all'interno del canale PalmOS

la croce al centro della mappa mostra la posizione dell'utente visto dall'alto e che guarda verso la parte superiore dello schermo.

Nello screenshot sopra riportato si vede che un altro utente è presente nel campo visivo ed è collocato dietro ed a destra dell'utente del palmare.

Per girarsi in direzione del visitatore basta premere qualche volta il tasto “svolta a destra”, ad ogni click si noterà una rotazione di 45° verso destra, cioè una rotazione apparente dell'avatar dell'altro utente in senso antiorario.

Per identificare un utente basta cliccare con la penna del palmare sul rettangolo che lo identifica, il nome dell'utente si sposta con l'avatar corrispondente permettendo di seguire i movimenti di un singolo utente anche all'interno della folla.

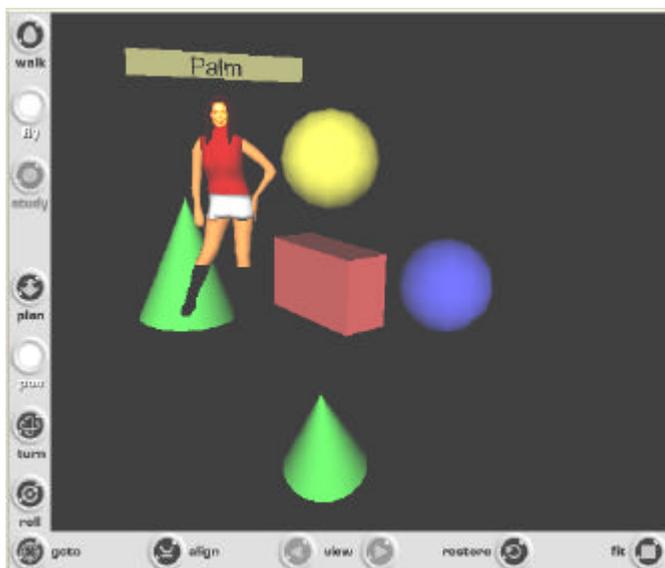
Per cambiare l'utente selezionato basta cliccare sul nuovo utente, mentre per deselectionarlo è sufficiente cliccare in una zona libera della mappa.

Nel caso in cui l'utente selezionato cambi nome, il nome visualizzato sopra il suo avatar cambia automaticamente senza bisogno di rifezionare l'utente.



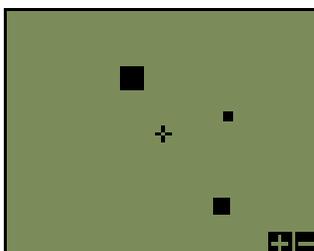
**Fig 7-36: La rappresentazione di un utente collegato attraverso un altro canale. La croce in centro rappresenta la posizione corrente dell'utente del canale Palm.**

Contemporaneamente l'utente collegato al canale tridimensionale (attraverso un PC dotato di browser e plug-in VRML) vede l'avatar selezionato da palmare girarsi nella sua direzione:



**Fig 7-37: La rappresentazione dello stesso spazio condiviso all'interno del canale dotato di rappresentazione tridimensionale**

La terza dimensione, cioè l'altezza relativa rispetto a quella dell'utente, viene rappresentata dalla grandezza del rettangolo che rappresenta ogni utente. Più l'utente è in alto (e quindi vicino alla telecamera, che inquadra la scena dall'alto in basso) più grande apparirà il suo avatar. Esistono tre diverse grandezze in cui si può presentare un avatar, a seconda dell'altezza raggiunta nello spazio visibile:



**Fig 7-38: Più utenti presenti nello spazio condiviso, che navigano su quote diverse**

Premendo i tasti “+” e “-” sulla mappa si aumenta o si diminuisce lo zoom, questo si concretizza in un allontanamento o un avvicinamento apparente degli avatar, occorre

però notare che il cambio di scala avviene anche lungo l'asse Y, questo comporta a volte il passaggio di un determinato utente da una zona di altezza all'altra ed il conseguente cambiamento della dimensione dell'avatar o addirittura la sua scomparsa nel caso in cui risulti troppo in alto o troppo in basso per il nuovo scope.

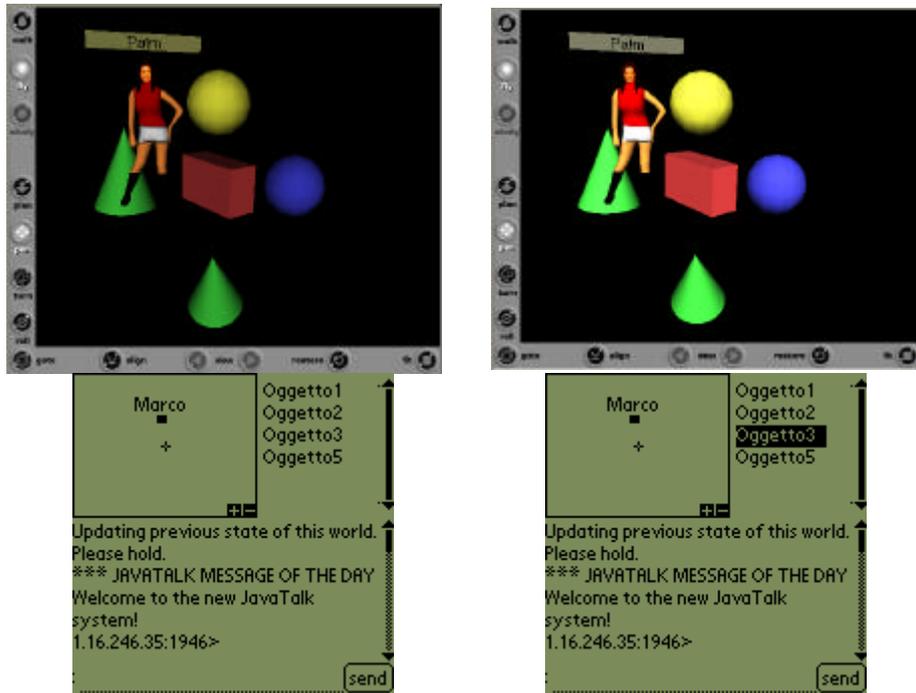
I messaggi di chat vengono aggiornati automaticamente "in tempo reale" sulla textBox che occupa la metà inferiore dello schermo, l'utente può utilizzare la scrollbar per visualizzare tutto lo storico della conversazione, ma quando arriva un nuovo messaggio il testo viene riportato automaticamente in fondo in modo da renderlo visibile.

Per dire una frase in chat basta scriverla utilizzando graffiti e premere il pulsante "send". Si noti che, dato che esiste un unico componente in cui si può inserire del testo, questo mantiene sempre il focus e non è necessario selezionarlo prima di incominciare l'inserimento del testo:

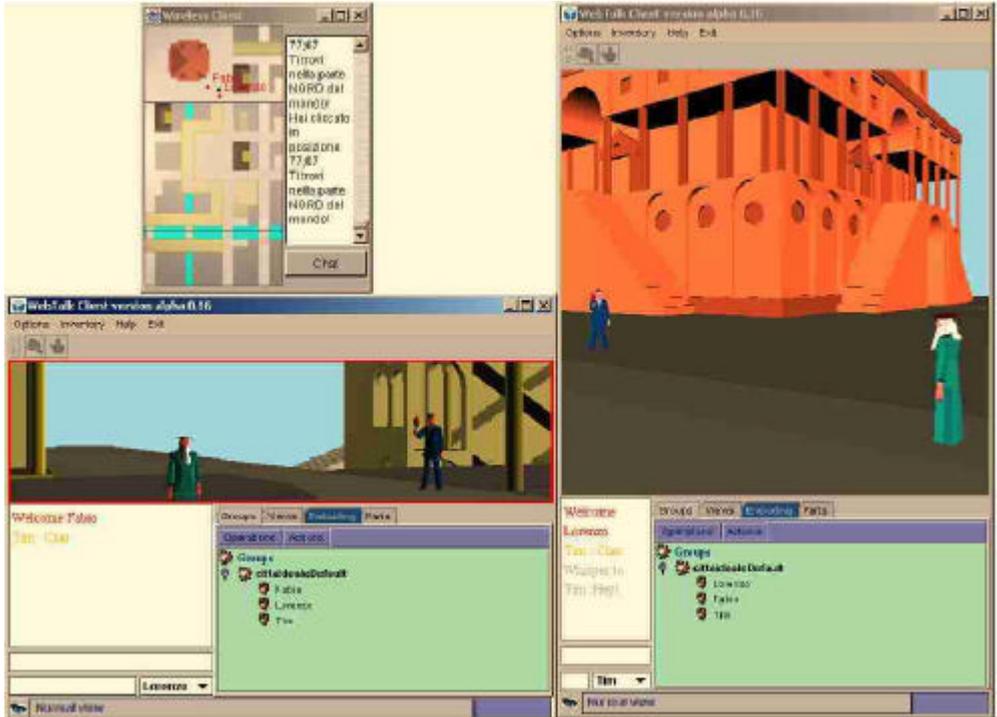


**Fig 7-39: Interazione multicanale utilizzando un classico scambio di messaggi testuali, lato canale Palm.**

L'interazione con gli oggetti condivisi è resa possibile grazie all'apposita lista degli oggetti posta alla destra della mappa. Ogni volta che un utente attiva un oggetto cliccandoci sopra, il nome dell'oggetto viene evidenziato nella lista e lo rimane fino a quando l'utente non rilascia il tasto del mouse; similmente l'utente del palmare può inviare una sequenza di comandi di attivazione e disattivazione dell'oggetto semplicemente cliccando sul nome dell'oggetto nella lista. Nell'esempio è stato attivato un oggetto a cui è associata, nel codice VRML, l'accensione di una luce nel mondo 3D. In questo caso l'utente su PC osserva l'evento di attivazione come l'accensione ed il successivo spegnimento della luce, mentre l'utente del palmare lo èerceòosce come la selezione e la successiva deselezione del nome dell'oggetto dall'elenco.



**Fig 7-40:** Una collaborazione multicanale in atto tra un utilizzatore del canale Palm e un utilizzatore di canale dotato di rappresentazione tridimensionale: l'utente Palm si trova in prossimità di un oggetto cliccabile, l'utente 3D ("Marco") si trova di fronte a lui. Cliccando su **Oggetto3** nella lista posta a destra, l'utente Palm esegue una attivazione dell'oggetto, che causa l'attivazione di una azione nella rappresentazione 3D (in particolare, all'oggetto è stato associata l'accensione di una luce). L'utente del canale 3D percepisce l'azione generata a partire dal canale Palm. Nel frattempo la conversazione tra gli utenti può avvenire nell'area di messaggi sottostante.



**Fig 7-41:** Una collaborazione multicanale utilizzando un *port* del canale palmare sotto WindowsCE e iPaq. L'applicazione caricata è La Città Ideale di Leonardo da Vinci. Si può osservare una interazione tra due utenti che utilizzano un canale dotato del client con Scene Builder Java 3D ed un client collegato mediante il palmare. Il palmare offre una visualizzazione simile a quella per PalmOS, con in più una rappresentazione grafica 2D della geometria in cui avviene la collaborazione, anche se questa mappa non rende vincolanti i movimenti dell'avatar del palmare.

## 8 Conclusioni e Sviluppi Futuri

Lo sviluppo di un sistema collaborativo multicanale con visualizzazione tridimensionale richiede una attenta analisi concettuale a vari livelli, ed è di difficile implementazione tecnica.

Dal punto di vista concettuale, occorre comprendere i seguenti punti:

- quale è la natura della collaborazione che si vuole offrire e come organizzarla: a questo punto abbiamo risposto con la teoria della percezione dello spazio condiviso e la teoria delle metafore cooperative
- quale è il modo con cui rappresentare lo spazio di lavoro condiviso: per questo abbiamo offerto come linee guida le metafore visive
- come far corrispondere contenuti bidimensionali (siti web) con un meccanismo di accesso collaborativo tridimensionale, attraverso la configurazione con il formalismo CML.
- come analizzare le proprietà dei diversi canali che si desidera interconnettere in un sistema collaborativo multicanale: per questo abbiamo offerto un'analisi utilizzando la "pila della multicanalità"

Questi quattro strumenti concettuali rispondono alla necessità di esplorare concettualmente il problema e l'applicazione, e forniscono un metodo definito di supporto alla progettazione di applicazioni che si muovono in questo ambito. Dalle esperienze raccolte nell'uso di questi modelli, possiamo sostenere che la guida che essi offrono è valida e adatta a poggiarvi sopra la realizzazione di applicazioni effettive.

Dal punto di vista implementativo, i nodi caratteristici da affrontare sono:

- la realizzazione di un renderer che sia in grado di sostenere il tracciamento in tempo reale degli aggiornamenti della rappresentazione: a questo problema abbiamo risposto proponendo l'architettura di Scene Builder, organizzata intorno ai concetti di Grafo di Scena e Behavior. Abbiamo tuttavia constatato che per risolvere problemi di natura prettamente prestazionale occorrono politiche di rendering mirate, meglio ottenibili mediante altre tecnologie, ed abbiamo verificato come il perseguimento di queste politiche offra tangibili vantaggi
- la realizzazione di un gestore dello stato condiviso che consenta di sostenere l'intercomunicazione tra gli utenti. Abbiamo riferito come in letteratura esistano diverse situazione tecniche al riguardo, tuttavia abbiamo aggiunto l'aspetto della multicanalità, e come sia necessario, per tenerne conto, organizzare dei punti di aggancio appositi, talvolta specifici per ciascun canale, al sistema di condivisione dello stato
- la constatazione che questi sistemi sono di difficile utilizzo per l'utente finale. Essi sono utilizzati con successo quando sono rivolti all'uso o all'addestramento da parte di utenti specializzati, e per un compito complesso (simulatori, piattaforme di addestramento per macchinari, etc.), ma sono molto

meno compresi da parte di utenti normali, che non hanno la cultura di questo tipo di applicazioni. Una risposta a questo problema è il concetto di Collaborazione Assistita, e la nostra proposta è la realizzazione di opportuni agenti atti alla guida e alla collaborazione con gli utenti effettivi

- il problema di rappresentare stati condivisi, modellati nelle tre dimensioni, su canali alternativi. Si è detto come sia sostanzialmente necessario operare delle scelte di rappresentazione diverse, forzatamente limitate dalle caratteristiche del particolare canale. Abbiamo portato due esempi applicativi, realizzati attorno a due prototipi, uno su un canale WAP e uno su un canale PDA dotato di accesso alla rete in larga banda.

E' probabile che il futuro di questi sistemi sia in rapida crescita ed evoluzione: tutti gli indicatori tecnologici lo fanno desumere, e la coscienza da parte del grande pubblico nell'uso di questi sistemi cresce, grazie alle varie implementazioni che cominciano ad esserne offerte. Occorre cogliere l'opportunità, sempre più concreta, che questi sistemi offrono, chiarendo fin da subito che a monte degli aspetti strettamente implementativi è necessario un modello concettuale per organizzare l'offerta dei contenuti e i meccanismi con cui si desiderano che siano fruiti.

L'aspetto della multicanalità relativo a questi sistemi, che ad oggi sembra un campo di indagine prematuro, si concretizzerà molto rapidamente: è sufficiente osservare, nel giro di pochi mesi, il balzo qualitativo compiuto dalle funzionalità offerte da palmari e cellulari. E vi è ancora spazio di miglioramento, anche per quanto riguarda la larghezza di banda di connessione in prospettiva G3.

Affrontare da oggi il problema consente di averne una solida comprensione nel momento in cui l'ultimo ostacolo, quello tecnologico, risulterà fondamentalmente superato, e consentirà la realizzazione di applicazioni che non siano solamente prototipali.

*Milano, Dicembre 2002*

## 9 Ringraziamenti

Il lavoro della presente ricerca è il frutto degli sforzi, dell'impegno e della passione di molte persone, che hanno lavorato per alcuni anni. A tutti invariabilmente, ai miei docenti, ai miei collaboratori, ed ai miei studenti, va un grazie sentito per l'entusiasmo con cui hanno condiviso le molte ore di lavoro, di discussione, e di sviluppo.

Grazie a mia moglie Olga, che mi ha sopportato con infinita pazienza nelle ore finali della stesura di questo lavoro, e mi ha sostenuto con il suo amore. Grazie a mio padre e mia madre, costante e discreto supporto nella mia vita.

Un grazie ai gruppi di lavoro che hanno lavorato e studiato con me:

**WebTalk-I:**

Paolo Loiudice  
Fabio Biscaro

**Leonardo Virtuale:**

Paolo Sena  
Giovanni Maranesi  
Mario Arrigoni Neri  
Marco Zanti

**WebTalk-II Scene****Builder:**

Matteo Foccoli  
Davide Affaticati

**Configuratore 3D:**

Filippo Naggi  
Roberto Ciatti

**WebTalk-II Server:**

Matteo Arru  
Ferdinando Carella

**Net-Vivos:**

Mauro Marchetto

**Client Multicanale:**

Federica Mega  
Fabio Lorenzo  
Mario Arrigoni Neri

**Collaborazione Assistita:**

Piergiovanni Santini

**Città Ideale:**

Giuseppe Neri  
Fabrizio Ferrari

**Museo Virtuale  
dell'Informatica:**

Giovanni Beltrami

**Santa Tecla:**

Massimiliano Pacchiani  
Stefano Buora

**Kernel OpenGL:**

Marco Ferra  
Paolo Costa  
Matteo Migliavacca  
Liliana Nenov

**MultiLOD:**

Fabio Rasia  
Luca Ferrara  
Andrea Sala

**Clipping:**

Paolo Banfi  
Alberto Piantanida

**Metafore Cooperative:**

Alessandro Rossi

Fabio Soverati

**Collision Detection:**

Gabriele Galantucci  
Andrea Galli  
Alessandro Magistroni

**WebBoard Wap****Converter:**

Roberto Cipriani

## 10 Bibliografia

### Publicazioni Personali

- [1] Paolini P., Barbieri T., et al. - "Visiting a Museum Together: how to share a visit to a virtual world" in *JASIS Journal of the American Society for Information Science Wiley&Sons, Ottobre 99*
- [2] Barbieri T. - "Reconstructing Leonardo's Ideal City – from handwritten codexes to WebTalk-II: a 3D Collaborative Virtual Environment" in *Proceedings ACM/SIGGRAPH VAST 2001, Athens, Greece, November 2001*
- [3] Barbieri T., Garzotto F., et al – "From Dust to Stardust: a Collaborative Virtual Computer Science Museum", in *Proceedings Short Papers Ichim 01, Milano, Italy, September 2001*
- [4] Barbieri T., Paolini P.– "Cooperation Metaphors for Virtual Museums", in *Proceedings Museums & Web 2001, Seattle, USA, March 2001*
- [5] Barbieri T. – "Networked Virtual Environments for the Web: The WebTalk-I and WebTalk-II Architectures", in *Proceedings IEEE for Computer Multimedia & Expo 2000 (ICME) , New York, USA, July 2000*
- [6] Barbieri T., Paolini P. – "Cooperative Visits to WWW Museum Sites a Year Later: Evaluating the effect", in *Proceeding Museums&Web2000, Minneapolis (USA), April 2000*
- [7] Barbieri T., Paolini P., Alonzo F. – "Cooperative Visits to 3D Virtual Museums", in *Proceedings International Conference for Cultural Heritage & MEDICI Day, Milano, Sett. 99*
- [8] Barbieri T., Paolini P. – "WebTalk: a 3DCollaborative Environment to Access the Web" in *Proceedings EUROGRAPHICS '99, Short Papers pg. 111-113, Settembre 99*
- [9] Paolini P., Barbieri T., et al. – "Visiting a Museum Together: how to share a visit to a virtual world", in *Proceedings Museums&Web '99, New Orleans (USA), Marzo 99, pg. 27 e segg.*

### Tesi di Laurea

- [10] Barbieri T. - *WebTalk-II: un'infrastruttura per la cooperazione sul web (Tesi di Laurea, Politecnico di Milano, 1999)*
- [11] Loiudice P. - *Cooperazione nella visita ad un museo virtuale (Tesi di Laurea, Politecnico di Milano, 1999)*

- [12] Ciatti R., Naggi F. - *Progettazione e Generazione di Ambienti Virtuali Condivisi: WebTalk-II* (Tesi di Laurea, Politecnico di Milano, 2000)
- [13] Foccoli M. - *WebTalk-II: Il modulo Scene Builder* (Tesina di Laurea, Politecnico di Milano, 2000)
- [14] Mega F. - *La Cooperazione Multicanale: WAP e WebTalk* (Tesi di Laurea, Politecnico di Milano, 2001)
- [15] Rossi A., Soverati F. - *Definizione di un modello per la cooperazione in ambienti virtuali, integrabile in WebTalk-II* (Tesi di Laurea, Politecnico di Milano, 2001)
- [16] Santini P. - *Agenti Autonomi nei Mondi Virtuali Cooperativi* (Tesi di Laurea, Politecnico di Milano, 2002)
- [17] Affaticati D. - *Sviluppo del Modello a Behavior e della Virtual Bag per WebTalk-II* (Tesina di Laurea, Politecnico di Milano, 2002)
- [18] Lorenzo F. - *Wireless Client: un'applicazione per la convergenza tra cooperazione e multicanalità* (Tesi di Laurea, Politecnico di Milano, 2002)
- [19] Arru M., Carella F. - *WebTalk-II: componenti server* (Tesi di Laurea, Politecnico di Milano, 2002)

### **Articoli di Altri sulla Nostra Ricerca**

- [20] *The New York Times On The Web*, Matthew Mirapaul – “At This Virtual Museum, You Can Bring a Date”, Giugno 99
- [21] *Java Industry Connection*, Matthew Mirapaul – “At This Virtual Museum, You Can Bring a Date” -  
<http://industry.java.sun.com/javanews/stories/story2/0,1072,16396,00.html> (ancora online)
- [22] *PC Professionale* (Mondadori), Valeria Camagni – “Con la tecnologia WebTalk navigare nel Web diventa un'attività sociale”, apparso su *PC Professionale* di Giugno 1999, pg. 246 e sgg.
- [23] *Internet Gazeta Wyborcza*, Dorota Gut, *Latajvce Machiny Leonardo* (“Le Macchine Volanti di Leonardo”), Ottobre 99
- [24] *Corriere della Sera*, Andrea Lawendel 17 Settembre 2001, pagina “MultiMedia” – “Come Ricreare al Computer la Città Ideale di Leonardo”, pag. 21  
([http://hoc.elet.polimi.it/barbieri/article\\_ideal\\_.jpg](http://hoc.elet.polimi.it/barbieri/article_ideal_.jpg))

---

## Multicanalità in ambito Mobile

- [25] *WAP Architecture - Version 30- Apr- 1998 – Wireless Application Protocol Architecture Specification, ? Copyright Wireless Application Protocol Forum, Ltd, 1998, available at <http://www.wapforum.org>*
- [26] *Jin Jing, Abdelsalam (Sumi) Helal, Ahmed Elmagarmid – Client-Server Computing in Mobile Environments – ACM Computing Surveys, Vol. 31, No. 2, June 1999*
- [27] *Wireless Application Protocol White Paper - Wireless Internet Today, October 1999, available at Wap Forum Web Site: <http://www.wapforum.org>*
- [28] *Martin Lippert, Henning Wolf, Heinz Züllighoven – Domain Services for Multichannel Application Software - in Proceedings of the 34<sup>th</sup> Hawaii International Conference System Sciences , 2001*
- [29] *Dillon A., Richardson J., McKnight C. –The effect of display size and text splitting on reading lengthy text from the screen -Behaviour and Information Technology, 9(3) (1990)*
- [30] *Norliza Mohd-Nasir, Kevin Boone, George Buchanan - Improving Web Interaction on Small Displays - in Proceedings of the 8<sup>th</sup> International World Wide Web Conference, Toronto, Canada, May 11-14 1999*
- [31] *S. Kristoffersen, F. Ljungberg – Your Mobile Computer is a stationary computer - in Online Proceedings of the ACM Workshop on Handheld CSCW, Seattle, Washington, November 1998*
- [32] *J. Perkins - Handscape Project Description – An investigation of Wireless Technology in the Museum Community – 30 Settembre 2000, available at CIMI Web Site: <http://www.cimi.org>*
- [33] *Henry Chang, Carl Tait, Norman Cohen, Moshe Shapiro, Steve Mastrianni - Web Browsing in a Wireless Environment: Disconnected and Asynchronous Operation in ARTour Web Express - in Proceedings of the 3th annual international on Mobile computing and networking, Budapest, Hungary, 1997*
- [34] *Philip K. McKinley, Ji Li – Pocket Pavilion: a synchronous collaborative browsing application for wireless handheld computers - in Proceedings of the 2000 IEEE International Conference on Multimedia and Expo, 30 Luglio – 2 Agosto 2000*
- [35] *Magnus Bergquist, Henrik Fagrell, Peter Ljungstrand, Marius Storsten – Designing for Informal mobile Cooperation - in Online Proceedings of the ACM Workshop on Handheld CSCW, Seattle, Washington, November 1998*
- [36] *Gerd Kortuem - Some Issues in the Design of User- Interface for Collaborative Wearable Computers - in Online Proceedings of the ACM Workshop on Handheld CSCW, Seattle, Washington, November 1998*

- [37] Paul Luff, Christian Heath - *Mobility in Collaboration - in Online Proceedings of the ACM Workshop on Handheld CSCW, Seattle, Washington, November 1998*
- [38] William E. Ridde – *Palmtop Collaboration: Problems and A Few Solutions - in Online Proceedings of the ACM Workshop on Handheld CSCW, Seattle, Washington, November 1998*
- [39] Albrecht Schmidt, Markus Lauff, Michael Beigl - *Handheld CSCW - in Online Proceedings of the ACM Workshop on Handheld CSCW, Seattle, Washington, November 1998*
- [40] James A.Landay, Richard C.Davis, Victor Chen, Jonathan Huang, Rebecca B.Lee, Francio Li, James Lin, Charles B.Morrey III, Ben Schleimer - *NotePals: Sharing and Synchronizing Handwritten Notes with Multimedia Documents - Proceeding of the CHI 99 conference on Human factors in computing systems : the CHI is the limit , 1999*
- [41] Brad A.Myers, Herb Stiel e Robert Gargiulo “*Collaboration Using Multiple PDAs Connected to PC*” *Proceedings CSCW’98: ACM Conference on Computer-Supported Cooperative Work, November 14-18, 1998, Seattle, WA. pp. 285-294.*

## Gestione dei contenuti

- [42] Garzotto F., Paolini P., Schwabe D. - *HDM - A Model Based Approach to Hypermedia Application Design - ACM Transactions on Information Systems, Vol.11,N.1,Jan. 1993*
- [43] Garzotto F. Mainetti L., Paolini P.- *Navigation in Hypermedia Applications: Modeling and Semantics, Journal of Organizational Computing, Vol. 6, N. 3, 1996.*
- [44] Garzotto F., Mainetti L. Paolini P.– *Navigation Patterns in Museum Hypermedia- ICHIM’95 – International Conference on Hypermedia for Museums, S. Diego (CA) Oct. 1996*
- [45] Garzotto F., Matera M., Paolini P. – *To use or not to use? Evaluating Usability of Museum Web Site – Int. Conf. On Museum and Web, Toronto , Canada, April 1998.*
- [46] Cap. 13: *Tools for Designer*". In *Hypertext/Hypermedia Handbook*, Berk, E. and Devlin, J. (Eds.), McGraw Hill, 1991 **Garzotto, F., Paolini P., Schwabe D., and Berstein M.**
- [47] <sup>1</sup> Mario A. Bochicchio, Roberto Paiano, Paolo Paolini: *JWeb: An HDM Environment for Fast Development of Web Applications. ICMCS, Vol. 2 1999: 809-813*

## Collaborazione e Ambienti Virtuali Collaborativi

- [48] S.Benford, D.Snowdon, C.Brown, G.Reynard and R.Ingram – *The populated Web: Browsing, Searching, and Inhabiting the WWW Using Collaborative Virtual*

- 
- Environments – Interact '97- the 6<sup>th</sup> IFIP Conference on Human-Computer Interaction incorporating OZCHI97 and APCHI97, 14-18, July, 1997, Sydney, Australia*
- [49] G.Sidler, A.Scott, H. Wolf – Collaborative Browsing in the World Wide Web – Proceedings of the 8<sup>th</sup> Joint European Networking Conference, Edinburgh, May 12.-15. 1997, available at <http://www.terena.nl>
- [50] Wolfgang Appelt - WWW Based Collaboration with the BSCW System - in Proceedings of the International of SOFSEM '99, Springer Lecture Notes in Computer Science 1725, pages 66-78, Milovy (Czech Republic), November 26- December 4 1999
- [51] Musse S., Babski C., Capin T, Thalmann D., Crowd Modelling in Collaborative Virtual Enviroments, proceedings of VRST98, Taipei Taiwan, 1998.
- [52] Zyda M., Singhal S., Networked Virtual Enviroments, ACM Press, NewYork, 1999.
- [53] Greenberg S., Roseman M., Using a Room Metaphor to Ease Transition in Groupware, Research report 98/611/02, Dept. of Computer Science, University of Calgary, Calgary, Alberta, Canada, 1998.
- [54] Frecon E., Nou A., Building Distributed Virtual Enviroments to Support Collaborative Work, Proceedings of VRST98, Taipei Taiwan 1998.
- [55] Schuckmann C., Schummer J., SeitzP., Modeling Collaboration using Shared Objects, Proceedings of GROU99, Phoenix USA, 1999.
- [56] Cerulli Cristina – Exploiting the potential of 3D navigable virtual exhibition spaces CDROM version of the proceedings of the ACM Museums and the Web 99 conference, New Orleans, March 11-14, 1999
- [57] Gutwin C. and Greenberg S. – A Framework of Awareness for Small Groups in Shared-Workspace Groupware. Thecnical Report 99-1, Departement of Computer Science, University of Saskatchewan, Canada, 1999.
- [58] Gaver W. – Sound Support for Collaboration , Proceedings of the second European Conference on Computer Supported Cooperative Work, 1991.
- [59] Broll W., Populating the Internet: Supporting Multiple Users and Shared Applications with VRML, Proceedings of VRML97, Monterey USA, 1997.
- [60] Campbell D., Design in Virtual Enviroments Using Architectural Metaphors: a HIT La Gallery. March Dissertation, University of Washington, 1996.
- [61] Charitos D., Defining existential space in virtual enviroments, Proceedings of Virtual Reality Worlds '96, Februry '96, IDG Magazines, Stuttgart, 1996.
- [62] Bridges H.A. and Charitos D., The architectural design in virtual enviroments, R. Junge (ed) CAAD Futures '97, Kluwer Academic Publishers, Dordrecht, 1997.

- [63] Patricia Wallace – *The Psychology of Internet* – 1999 Cambridge University Press.
- [64] Kristoffersen S., Ljumberg F., *An Empirical Study of How People Establish Interaction: Implications for CSCW Session Management Models*, in *Proceedings CHI99*, Pittsburgh USA, 1999.
- [65] Charitos D., Rutherford P., *Way of aiding navigation within VRML Worlds*, *Proceedings of the 6<sup>th</sup> EuroPLA Conference*, Edimburgh, 1997.
- [66] Pttie Maies and Leonard N. Foner – *Payng Attention to What’s Important: Using Focus of Attention to Imrove Unsupervised Learning- Proceeding from the Third International Conference on the Simulation of Adaptative Behaviour (SAB94)*, Brighton, England 1994.
- [67] M. Hearst, C. Karadi, *Cat-a-cone: An Interactive Interface for Specifying Searches and Viewing Retrieval Results using a Large Category Hierarchy*, *proceedings of 20<sup>th</sup> Annual International ACM/SIGIR Conference*, Philadelphia, PA, July 1997.
- [68] Pavel Curtis, "Mudding: Social Phenomena in Text-Based Virtual Realities", *Procedings of DIAC '92*
- [69] S. Benford, D. Snowdon, et al. *Visualising and Populating the Web: Collaborative Virtual Environments for Browsing, Searching and Inhabiting Workspace*, in *proceedings JENC8*
- [70] Cerulli, C. *Design of 3D navigable Virtual Exhibition Spaces*, in *proceedings Museum&Web 99*, New Orleans
- [71] Charitos, D., Rutherford, P., *Ways of aiding navigation within VRML Worlds*, *Proceedings of the 6<sup>th</sup> EuroPLA Conference*, Edimburgh, 1997
- [72] Campbell, D., *Design in Virtual Environments Using Architectural Metaphors: a HIT La Gallery*. March Dissertation, University of Washington, 1996
- [73] Charitos, D., *Defining existential space in virtual environments*, *Proceedings of Virtual Reality Worlds '96*, February '96, IDG Magazines, Stuttgart, 1996
- [74] Greenalgh C. – *Analysing Awareness Management in Distributed Virtual Environments - in Proceedings 2nd Annual Workshop on System Aspects of Sharing a Virual Reality*, part of CVE98, Manchester UK June 1999
- [75] Endsley, M., *Toward a Theory of Situation Awareness in Dynamic Systems*, *Human Factors*, 1995
- [76] Gutwin C., Greenberg S., *A Descriptive Framework of Workspace Awareness for Real-Time Groupware*. *Computer Supported Cooperative Work*, Kluwer Academic Press, 2001

- 
- [77] Manny Vellon, Kirk Marple, Don Mitchell, Steven Drucker, "The Architecture of a Distributed Virtual Worlds System", *USENIX Conference Paper*, 2000
- [78] Benford S., Lennart E., Fahlén E. - *A Spatial Model on Interaction in Large Virtual Environments - in Proceedings of the Third European Conference on CSCW, Italy, 1993*
- [79] Bullok, Benford – *An access control framework for Multi-User Collaborative Environments – in proceedings ACM GROUP99*
- [80] Keith Edwards – *Awareness and collaboration through policies and roles – in proceedings ACM CSCW96*

## Gestione dello stato condiviso

- [81] Aronson, J. - *Dead Reckoning: Latency Hiding for Networked Games – Gamasutra, 1997.*
- [82] Bricken, William and Geoffrey Coco - *The VEOS Project. Technical Report, Seattle, WA: Human Interface Technology Laboratory, University of Washington, 1993.*
- [83] Brutzman - *Distributed Interactive Simulation DIS-Java-VRML Working Group - <http://www.web3D.org/WorkingGroups/vrtp/dis-java-vrml/dis-java-vrml.html>*
- [84] Brutzman D., Zyda M., Watsen K., Macedonia M. - *Virtual Reality Transfer Protocol (vrtp) rationale - Proceedings of the IEEE Sixth International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE97), Distributed System Aspects of Sharing Collaborative Enterprises (WETICE97), Distributed System Aspects of Sharing a Virtual Reality workshop. IEEE Computer Society, Cambridge (MA).*
- [85] Calvin, James, Alan Dickens, Bob Gaines, Paul Metzger, Dale Miller, and Dan Owen - *The SIMNET Virtual World Architecture - Proceedings of the IEEE Virtual Reality Annual International Symposium, 1993.*
- [86] Coco, Geoffrey P. - *The Virtual Environment Operating System: Derivation, Function, and Form, Master's Thesis, College of Engineering, University of Washington, 1993.*
- [87] Carlsson, Christer, and Olaf Hafsand - *DIVE: A Multi-User Virtual Reality System - Proceedings of the IEEE Virtual Reality Annual International Symposium, 1997.*
- [88] Codella, Christopher F., Reza Jalili, Lawrence Koved, and J. Bryan Lewis - *A Toolkit for Developing Multi-User, Distributed Virtual Environments, Proceedings of the IEEE Virtual Reality Annual International Symposium, 1993.*

- [89] Don Brutzman - *Graphics Internetworking: Bottlenecks and Breakthroughs - Chapter 4, Digital Illusions*, Clark Dodworth editor, Addison-Wesley, Reading Massachusetts, 1997 - <http://web.nps.navy.mil/~brutzman/vrml/breakthroughs.html>.
- [90] Eric Cronin, Burton Filstrup, Anthony Kurc - *A Distributed Multiplayer Game Server System - Electrical Engineering and Computer Science Department University of Michigan*, 2001.
- [91] Grant, Robert - *MultiVerse System README* - <ftp://ftp.u.washington.edu/public/virtual-worlds/multiverse/>.
- [92] Greenhalgh, Chris, and Steve Benford - *MASSIVE: a Collaborative Virtual Environment for Tele-conferencing - ACM Transactions on Computer-Human Interaction (ToCHI)*.
- [93] Greenhalgh, Chris - *Chris Greenhalgh - MASSIVE95* - <http://www.crg.cs.nott.ac.uk/~cmg/massive95.html>.
- [94] Grimsdale, C. - *dVS: Distributed Virtual Environment System - Division, Ltd*, 1990.
- [95] Ishii, Masahiro, Masanori Nakata, and Makoto Sato - *Networked SPIDAR: A Networked Virtual Environment with Visual, Auditory, and Haptic Interactions, Presence: Teleoperators and Virtual Environments*.
- [96] Kazman, Rick - *Making WAVES: On the Design of Architectures for Low-end Distributed Virtual Environments - Proceedings of the IEEE Virtual Reality Annual International Symposium*, 1993.
- [97] Macedonia, Michael, R. Michael J. Zyda, David R. Pratt, and Paul T. Barham - *Exploiting Reality with Multicast Groups: A Network Architecture for Large Scale Virtual Environments - Proceedings of IEEE Virtual Reality Annual International Symposium*, 1995.
- [98] Macedonia, M.R. and Zyda, M.J. - *A Taxonomy for Network Virtual Environments*, *IEEE Multimedia*, Vol. 4, No. 1, Jan-Mar 1997.
- [99] Macedonia, Michael, R. Michael J. Zyda, David R. Pratt, and Paul T. Barham -, *NPSNET: A Multi-Player 3D Virtual Environment Over the Internet, ACM SIGGRAPH 1995 Symposium on Interactive 3D Graphics*.