

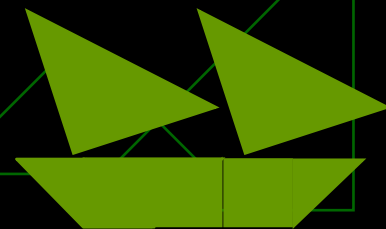
Intelligenza del software: inglobare la conoscenza musicale nei sistemi

Carmine Emanuele Cella
(cecily@libero.it)

- ◆ E' molto difficile trovare parametri per formalizzare la acquisizione della conoscenza
- ◆ Spesso il software si basa su conoscenze ben radicate
- ◆ Tale orientamento produce software "orientato al modello"
- ◆ Non è un caso che il modello dominante di software sia centrato sul concetto di partitura (score)
- ◆ Nel 20° secolo c'è stato uno spostamento dalla composizione basata su modelli ad una composizione "basata su regole"
- ◆ Tale approccio è ben indirizzabile nella computer music
- ◆ Ad esempio nei sistemi in tempo-reale la conoscenza del compositore è inserita nel sistema mediante azioni
- ◆ In tali sistemi però non esiste un feedback di conoscenza
- ◆ Spesso lo sviluppo compositivo è strettamente connesso allo sviluppo dello stesso sistema software

- ◆ Il principale problema non è di carattere fisico o di contesto, bensì è causato dalla difficoltà di analizzare il processo creativo
- ◆ Due approcci sono possibili: **BOTTOM-UP** e **TOP-DOWN**

- ◆ La creazione del software si sviluppa da esigenze compositive: ciò permette di sviluppare metodologie compositive sempre nuove e non solo basate su metodologie esistenti
il processo mentale del compositore crea un legame tra il livello sonoro e quello sintattico
- ◆ E' fondamentale l'organizzazione dei dati in un unità logiche superiori (oggetti sonori, gestures, eventi, ecc.)
- ◆ Un sistema che ingloba conoscenza obbliga ad un certo "pregiudizio" artistico mentre un programma "zero-knowlegde" risulta molto difficile da usare a causa della poca concettualità dei suoi elementi
- ◆ Gli aspetti chiave dell'approccio bottom-up sono il processo di automazione e la generalizzazione



- ◆ La creazione del software si sviluppa da una precisa idea/necessità:

ogni progetto è strettamente connesso all'attività di un compositore ed è difficilmente comunicabile

spesso un'idea compositiva è difficilmente formalizzabile (cfr. quantum leap)

L'output musicale è spesso tanto complesso da non poter essere predetto; il feedback acustico risulta indispensabile

- ◆ Gli aspetti chiave dell'approccio top-down sono la creazione di linguaggi ad alto livello che permettano però al compositore di operare a livelli anche più bassi, in modo da consentire una certa flessibilità nelle idee che strutturano il sistema

- ◆ I processi bottom-up e top-down, in realtà non sono lineari. Spesso sono discontinui o irrazionali o coinvolgono ciò che può essere chiamata “quantum leap” : un gap logico tra l’intuizione artistica e l’attuale concezione del software.
- ◆ L’uso “improprio” dei sistemi software si rivela a volte l’unico mezzo per comprenderne i limiti (noise-driven tests)
- ◆ Finchè i sistemi informatici non permetteranno la coesistenza di tutti i parametri che caratterizzano la conoscenza musicale (anche quelli irrazionali) difficilmente essi saranno soddisfacenti
- ◆ Il problema fondamentale dunque non è trovare nuovi algoritmi o codici per la costruzione di sistemi, bensì imparare come “campionare” il continuum di conoscenze all’interno della mente del compositore
- ◆ Al momento, il miglior metodo per raggiungere tale scopo è l’uso del feedback tra musicista e programmatore
- ◆ Tale compito di collaborazione è spesso definito “ingegneria della conoscenza” (O. Laske)

