

2. OPUSMODUS

Opusmodus è un sistema esperto per la composizione e l'analisi della musica sviluppato in Common Lisp. Opusmodus si compone di tre sezioni fondamentali:

Interfaccia grafica utente
Linguaggio OMN
Funzioni del sistema

2.1. Interfaccia grafica utente

Oltre a un menu classico contenente i comandi più comuni simili a molte applicazioni ampiamente utilizzate, Opusmodus offre cinque pannelli interattivi. I più significativi di questi sono:

- Il pannello “Compositore” (**Composer**), dove è scritto il testo-codice della composizione; qui il materiale musicale è descritto in linguaggio OMN e se ne può commentare il processo di elaborazione.
- Il pannello “Listener” che offre la possibilità di una valutazione interattiva delle singole espressioni (o dell'intero testo-codice) e permette di vederne i risultati di ritorno, evidenziandovi ogni possibile errore. Grazie a questo pannello è possibile verificare il proprio lavoro, passo dopo passo.
- Il pannello “Assistente” (**Assistant**) ove è possibile visualizzare istantaneamente ed interattivamente ogni singolo dettaglio del proprio lavoro in notazione tradizionale e verificare il risultato finale (la partitura), pronto per essere salvato, stampato o esportato in formato **MusicXML**. Il pannello “Assistente” può anche essere suddiviso, per visualizzare altri elementi tra cui: a) i grafici (plot) che forniscono la rappresentazione grafica di un certo materiale; b) il **MIDI-Player**, un vero MIDI sequencer dove si può ascoltare la partitura controllandola visivamente; c) la documentazione in linea in cui si possono anche provare i diversi esempi di ciascuna funzione.

2.2. Linguaggio OMN

(q)
(q c4)
(q c4 mp)
(q c4 mp trem)

La notazione OMN è uno script alfanumerico in grado di rappresentare l'intero vocabolario della musica (occidentale) in modo estremamente accurato, rispettandone ogni sfumatura. Tutti gli elementi della notazione musicale corrispondono a un simbolo specifico (o a un gruppo di simboli) del linguaggio OMN e viceversa. Dunque una notazione OMN è costituita da liste di simboli che rappresentano ogni parametro musicale. Queste liste si chiamano «forme OMN». Una forma OMN consente anche simboli abbreviati, indicanti estensioni e ripetizioni. Una forma OMN inoltre può essere strutturata su più livelli, ossia può essere formata da più sotto-liste.

2.2.1. I quattro elementi

Prerequisito essenziale per comporre musica con l'ausilio del computer è che le idee musicali di una composizione devono essere comunicate al computer nel modo più corretto e completo possibile: a tale scopo OMN mette a disposizione una notazione musicale esattamente equipollente alla notazione tradizionale su foglio. La domanda immediata quindi è: come funziona OMN in modo che un compositore possa comunicare al computer le sue idee musicali? Come la notazione tradizionale, OMN esprime unità musicali in termini di durate, altezze, dinamiche e articolazioni. I quattro elementi OMN, che indicano un «event» (unità musicale di OMN) sono:

1. durata, es. q (quarto)
2. altezza, es. c4 (do centrale)
3. velocity (dinamica), es. mp
4. articolazione, es. trem(olo)

disposti proprio in questo preciso ordine.

'(q c4 mp trem)



Ovviamente, le unità musicali possono raggrupparsi e organizzarsi in sequenze:

```
'((s a4 d5 fs4 d5 g4 d5)
  (s a4 d5 fs4 d5 g4 d5)
  (s a4 d5 cs5 b4 a4 g4)
  (s fs4 d4 e4 cs4 e d4))
```

Come per il Lisp, per mettere in moto OMN occorre formulare un'espressione. Un'espressione è una lista, che inizia con una parentesi aperta, è seguita da un certo numero di simboli e infine si chiude con una parentesi chiusa.

```
(gen-retrograde '(s a4 d5 fs4 d5 g4 d5))
=> (s d5 g4 d5 fs4 d5 a4)
```

L'espressione sopra è una lista in cui il primo elemento è il nome di una funzione. Il resto della lista sono gli argomenti (i dati) a cui viene applicata la funzione.

Poiché il Lisp proverà a valutare tutto ciò che scrivi, deve esserci un modo per dire a Lisp di considerare alcune espressioni solo come dati. Per informare il Lisp che si desidera che una lista venga considerata solo come "dati" bisogna «quotare» tale lista, premettendo alla parentesi aperta un apice "'":

```
'((s a4 d5 fs4 d5 g4 d5) (s a4 d5 fs4 d5 g4 d5)
  (s a4 d5 cs5 b4 a4 g4) (s fs4 d4 e4 cs4 e d4))
```

Proprio come accade nell'ordinaria scrittura di testi: nel caso in cui vogliamo dire che Parigi è la capitale della Francia, usiamo il termine senza virgolette, ma usiamo le virgolette di citazione quando vogliamo dire che la parola «Parigi» ha cinque lettere.

Quindi, se vuoi che Lisp veda (q c5 ff tr2) come dati, lascia che l'espressione sia preceduta da un apice:

```
'(q c5 ff tr2)
```

Proviamo l'espressione (gen-integer 12) con e senza virgolette e vediamo la differenza. Per farlo dobbiamo valutare la nostra espressione: posizionare il cursore dopo l'ultima parentesi di chiusura “)” e premere il tasto invio. La valutazione verrà visualizzata nel pannello **Listener**.

```
(gen-integer 12) ;restituisce un elenco di numeri da 0 a 12  
=> (0 1 2 3 4 5 6 7 8 9 10 11 12)
```

```
'(gen-integer 12) ;è una lista con 2 valori  
=> (gen-integer 12)
```

NB: il segno di freccia “=>” utilizzato in tutto il libro al pari che negli esempi della documentazione in linea non è computabile: sta solo a segnalare il risultato di una valutazione, il punto e virgola “;” introduce un commento e pertanto ciò che lo segue nella stessa riga non viene valutato.

A seguire illustriamo due funzioni fondamentali di Opusmodus e del Lisp.

setf

Se vuoi elaborare una sequenza di dati è fondamentale assegnare quella sequenza a una variabile: **setf** ci permette di farlo. Qui sotto **setf** assegna alla variabile `song` una sequenza di forme `omn`.

```
(setf  
 song  
 '( (3e gb6 bb6 db6 gb6 eb6 gb6 db6 gb6 bb5 db6 gb5 bb5)  
    (3e gb5 bb5 db5 gb5 eb5 gb5 db5 gb5 bb4 db5 gb4 bb4)  
    (3e db4 db5 ab4 db5 ab4 ab5 ab4 ab5 eb5 ab5 eb5 eb6)  
    (3e eb5 eb6 ab5 eb6 ab5 ab6 ab5 ab6 db6 ab6 db6 db7)))
```

Dopo l'assegnazione, si può usare `song` per riferirsi ai suoi contenuti.

La funzione **setf** consente assegnazioni multiple:

```
(setf prima '(a b c) seconda '(d e f) terza '(g h i))
```

list

La funzione **list** come dice il nome stesso consente la creazione di liste. Le liste possono avere qualsiasi lunghezza, quindi la funzione **list** accetta un numero indifferente di argomenti (sia elementi singoli che altre liste).

```
(list '(q c4 mp tr2)
      '(q c5 f fermata)
      '(q c5 ff tr2)
      '(q c6 fff fermata))
```

2.2.2. Le durate

Nel linguaggio OMN i valori di durata hanno la seguente corrispondenza:

8/4 d | 4/4 w | 2/4 h | 1/4 q | 1/8 e | 1/16 s | 1/32 t | 1/64 x | 1/128 u ||

premettendo il segno “-“ (meno) si ottengono gli equivalenti valori di pausa:

8/4 - | 4/4 - | 2/4 - | 1/4 - | 1/8 - | 1/16 - | 1/32 - | 1/64 - | 1/128 - ||

-d -w -h -q -e -s -t -x -u

in alternativa è sempre possibile sostituire le lettere dell’alfabeto con i corrispondenti valori frazionari:

8/4 → d	-8/4 → -d
4/4 → w	-4/4 → -w
2/4 → h	-2/4 → -h
1/4 → q	-1/4 → -q
1/8 → e	-1/8 → -e
1/16 → s	-1/16 → -s
1/32 → t	-1/32 → -t
1/64 → x	-1/64 → -x
1/128 → u	-1/128 → -u