Adaptive lookahead planning for performing music composition
Dimitri Bouche, Jean Bresson

To cite this version:
Dimitri Bouche, Jean Bresson. Adaptive lookahead planning for performing music composition. The 25th International Conference on Automated Planning and Scheduling (ICAPS), Jun 2015, Jerusalem, Israel. 2015. <hal-01163294>

HAL Id: hal-01163294
https://hal.archives-ouvertes.fr/hal-01163294
Submitted on 12 Jun 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Adaptive lookahead planning for performing music composition

Dimitri Bouche – Jean Bresson
UMR 9912 STMS IRCAM-CNRS-UPMC / Paris, France
{bouche,bresson}@ircam.fr

Context

• Creating music amounts to write (compose) and perform (live)
• Composition can last an indefinite time
• Performance has a timeline defined during composition

Adaptive lookahead (dynamic vs. static)

Dynamic object

Static object

likely to be modified while being rendered
will never change

Compositional processes that generate or modify musical structures can be integrated in musical objects, and occur during the performance

Planning of a musical object

 play-chord
 play-note
 note-on
 note-off

Dynamic lookahead extension

Plan « just in time » in a time-window

Possibility to extend this time-window

short anticipation
larger anticipation

Planning and scheduling operation scheme

Data structure

object to plan
translation

Environment & Compositional Processes

• Computer-music environment: react to user inputs
• Process tasks: non preemptible, best-effort
• Integrate results in the data structure
• Notify the planner about editions

Scheduler

• Synchronously render plans
• Query the planner for new plans after depletion
• Trigger compositional processes as « tasks »
• Time updates (graphics, callbacks etc.)

Planner

• Register playing objects
• Store time-windowed plans
• Extract new plans from the data structure according to queries or notifications

Example

Tasks triggering compositional processes

1 2 3: Results of compositional processes

EFFECTAC e ANR-13-JS02-0004-01: http://repmus.ircam.fr/effectace/
Download OpenMusic: http://repmus.ircam.fr/openmusic/