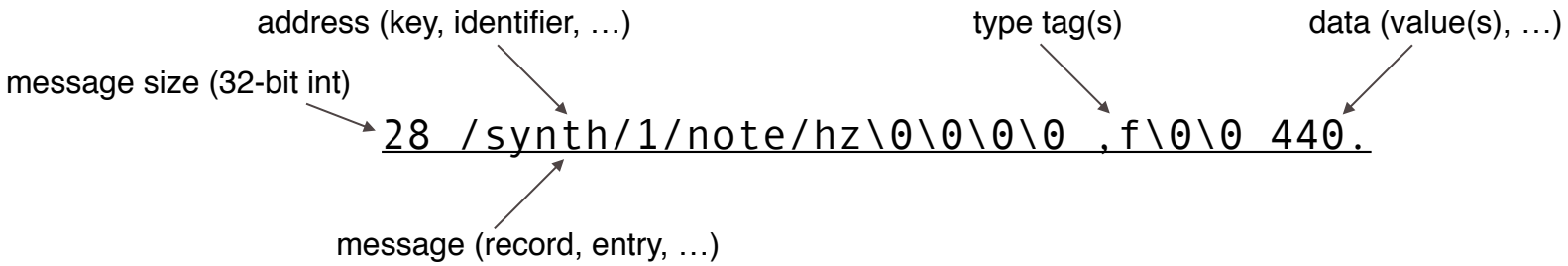
o.OM: Structured-Functional Communication between Computer Music Systems using OSC and Odot

Jean Bresson, John MacCallum, Adrian Freed
UC Berkeley — Center for New Music and Audio Technologies
IRCAM / UMR 9912 "STMS" CNRS — UPMC Sorbonne Universités

EFFICAC(e) — ANR-13-JS02-0004-01
Extended Frameworks for "In-time" Computer-Aided Composition

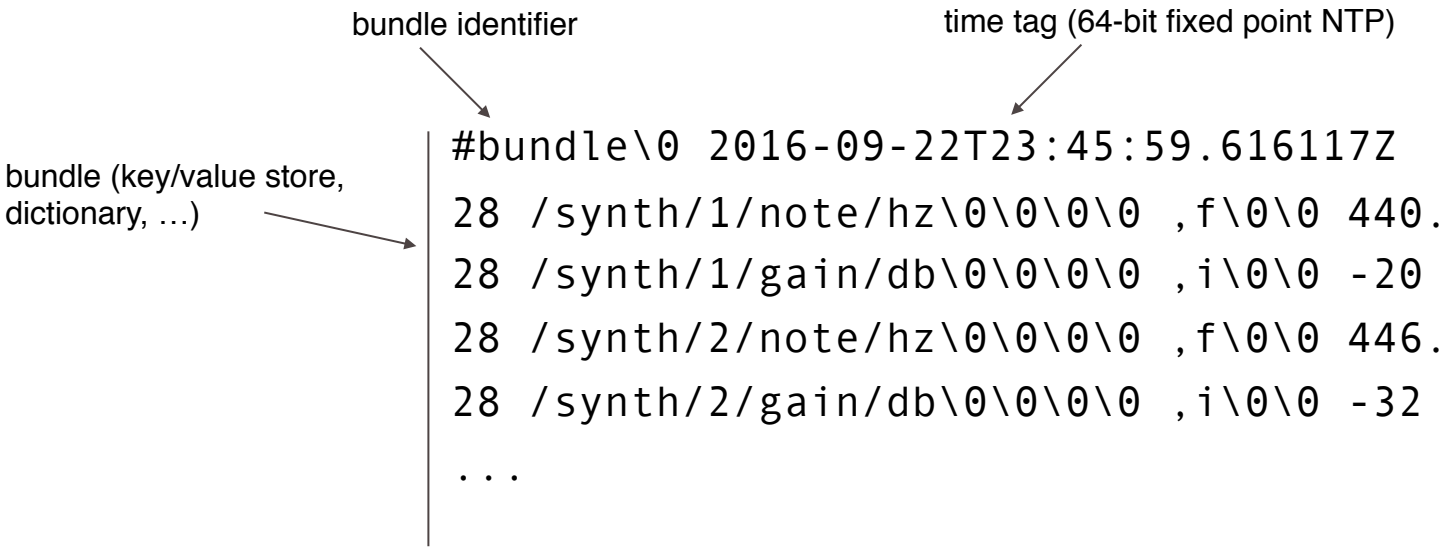
Open Sound Control (OSC)

- *lingua franca* for interactive media/arts programming
- binary encoding
- key/value store
- punctuation
- http://opensoundcontrol.org/spec-1_0



Open Sound Control (OSC)

- *lingua franca* for interactive media/arts programming
- binary encoding
- key/value store
- punctuation
- http://opensoundcontrol.org/spec-1_0



• M. Wright. **Open Sound Control: an enabling technology for musical networking.** Organised Sound, 10(3), 2005.

Open Sound Control (OSC)

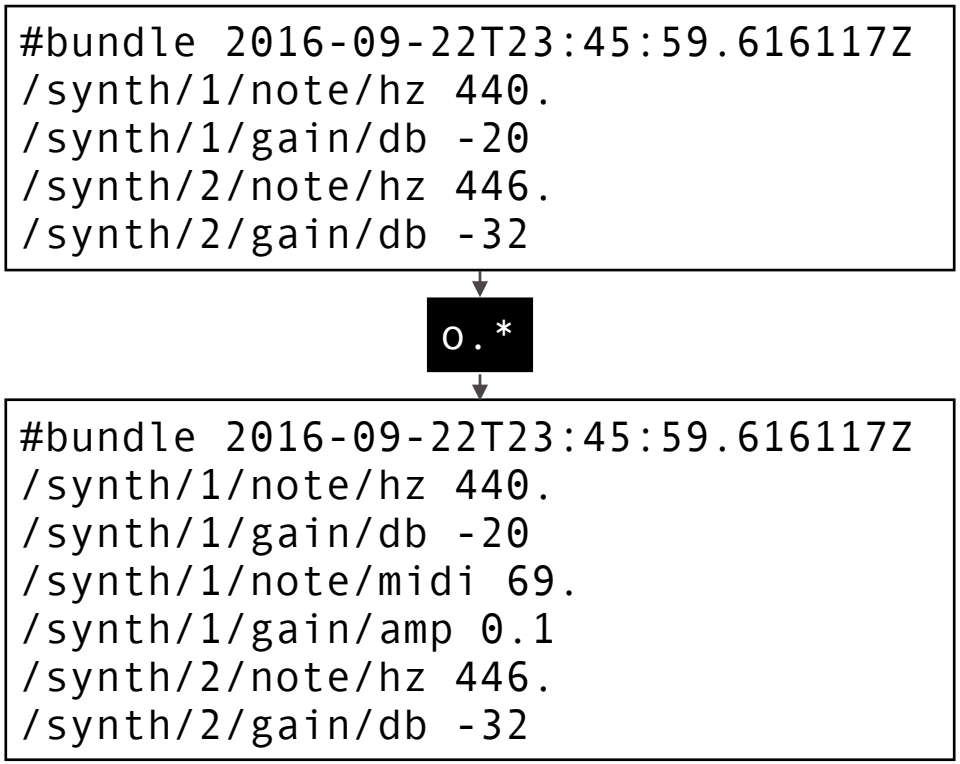
- *lingua franca* for interactive media/arts programming
- binary encoding
- key/value store
- punctuation
- http://opensoundcontrol.org/spec-1_0

```
/synth/*/note/hz 440.
/synth/[1-3]/note/hz 440.
/synth/{1,3,5}/note/hz 440.
```

- M. Wright. **Open Sound Control: an enabling technology for musical networking.** Organised Sound, 10(3), 2005.

o.*

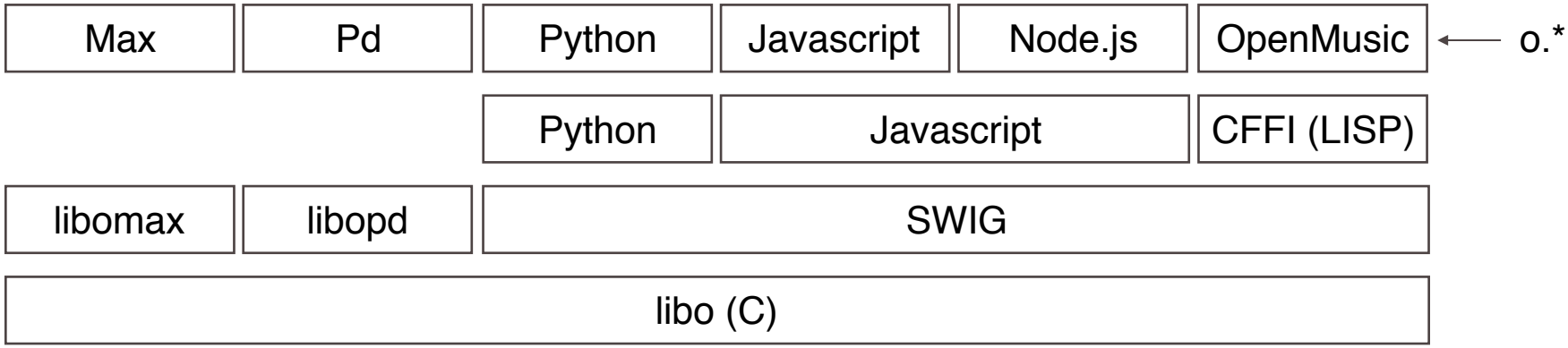
- superset of osc 1.1 (additional types)
- dynamic programming environment
- embedded in a host environment
- includes a small, lightweight expression language evaluator (o.expr)
- https://github.com/CNMAT/CNMAT-odot



- A. Freed, J. MacCallum, and A. Schmeder. **A Dynamic, Instance-Based, Object-Oriented Programming in Max/MSP using Open Sound Control Message Delegation.** In Proceedings of the International Computer Music Conference, Huddersfield, UK, 2011.

o.*

- superset of osc 1.1 (additional types)
- dynamic programming environment
- embedded in a host environment
- includes a small, lightweight expression language evaluator (o.expr)
- https://github.com/CNMAT/CNMAT-odot

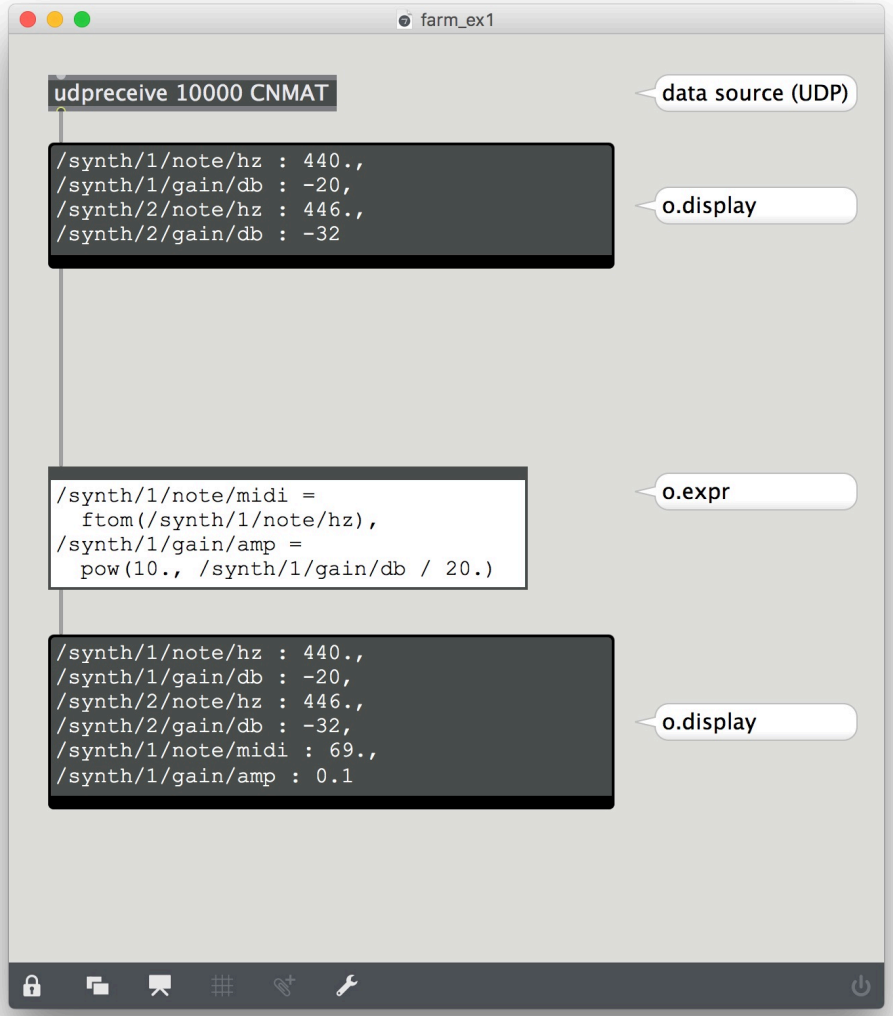


- A. Freed, J. MacCallum, and A. Schmeder. **A Dynamic, Instance-Based, Object-Oriented Programming in Max/MSP using Open Sound Control Message Delegation.** In Proceedings of the International Computer Music Conference, Huddersfield, UK, 2011.

–non-atomic
–loss of documentation
–loss of time stamp
–implicit type conversion

• A. Freed, J. MacCallum, and A. Schmeder. **A Dynamic, Instance-Based, Object-Oriented Programming in Max/MSP using Open Sound Control Message Delegation.** In Proceedings of the International Computer Music Conference, Huddersfield, UK, 2011.

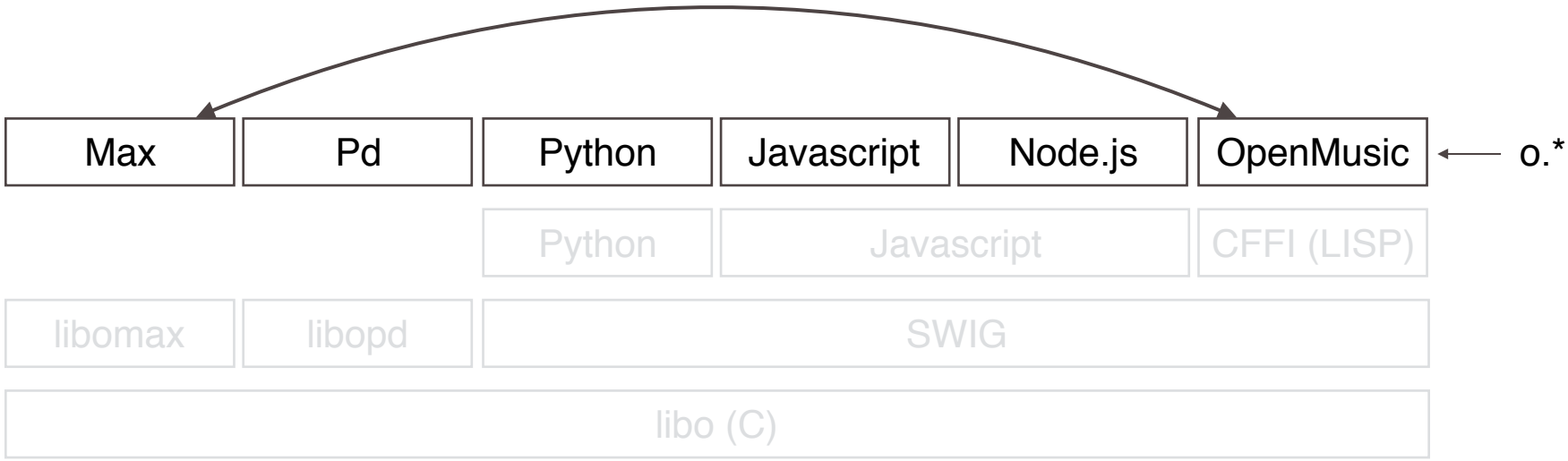
–atomic
–preserves documentation
–preserves time stamp
–no type conversion



• A. Freed, J. MacCallum, and A. Schmeder. **A Dynamic, Instance-Based, Object-Oriented Programming in Max/MSP using Open Sound Control Message Delegation.** In Proceedings of the International Computer Music Conference, Huddersfield, UK, 2011.

–decouples ‘what’ is computed from ‘where’ it is computed

• A. Freed, J. MacCallum, and A. Schmeder. **A Dynamic, Instance-Based, Object-Oriented Programming in Max/MSP using Open Sound Control Message Delegation.** In Proceedings of the International Computer Music Conference, Huddersfield, UK, 2011.



• A. Freed, J. MacCallum, and A. Schmeder. **A Dynamic, Instance-Based, Object-Oriented Programming in Max/MSP using Open Sound Control Message Delegation.** In Proceedings of the International Computer Music Conference, Huddersfield, UK, 2011.

- small, lightweight functional expression language
- function abstraction, and higher order functions
- no side effects

1.0:

- not designed
- brittle and awkward syntax
- multiple languages

2.0:

- designed :)
- robust syntax and semantics
- homoiconic

• A. Freed, J. MacCallum, and A. Schmeder. **A Dynamic, Instance-Based, Object-Oriented Programming in Max/MSP using Open Sound Control Message Delegation.** In Proceedings of the International Computer Music Conference, Huddersfield, UK, 2011.

OpenMusic: Visual programming environment for Computer-Aided Composition

http://repmus.ircam.fr/openmusic/

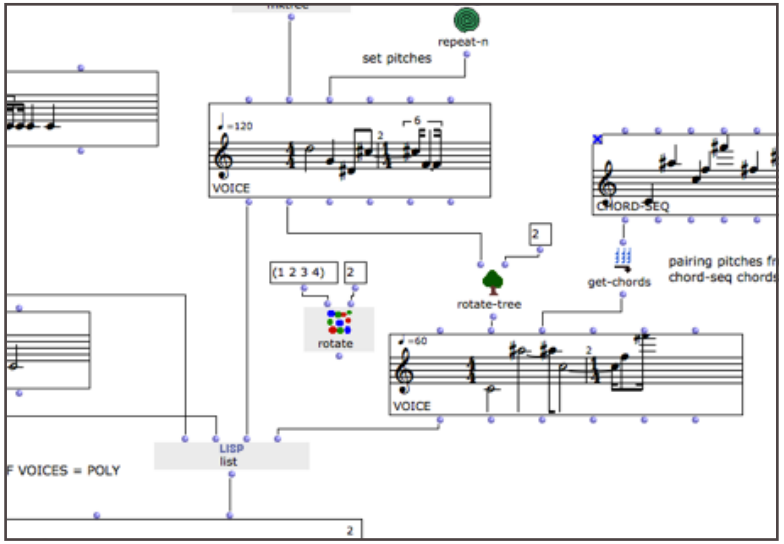
- Visual programming language built on top of LISP

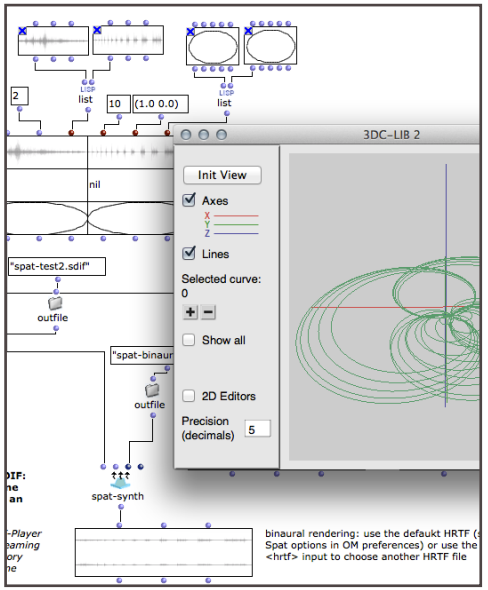
- Used by composers to implement compositional processes (generation/transformation of musical structures).

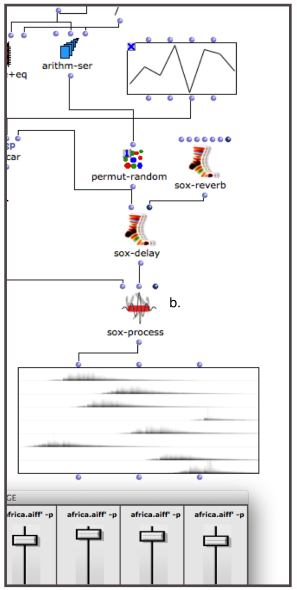
- Demand-driven execution

- Local state — Use graphical editors to visualise and edit input/intermediate/output data (scores, sounds, etc.)

- Specialised libraries for specific computing or musical approaches (chaos, probabilistic models, constraint programming, DSP, etc.)

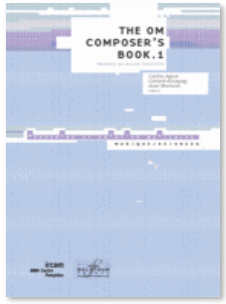


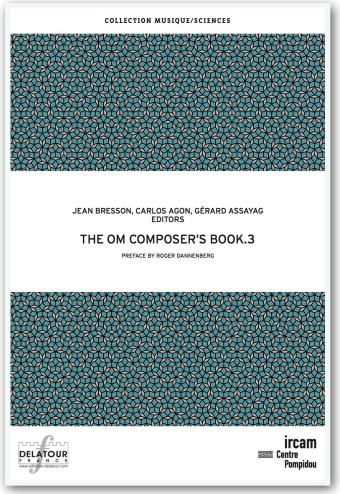




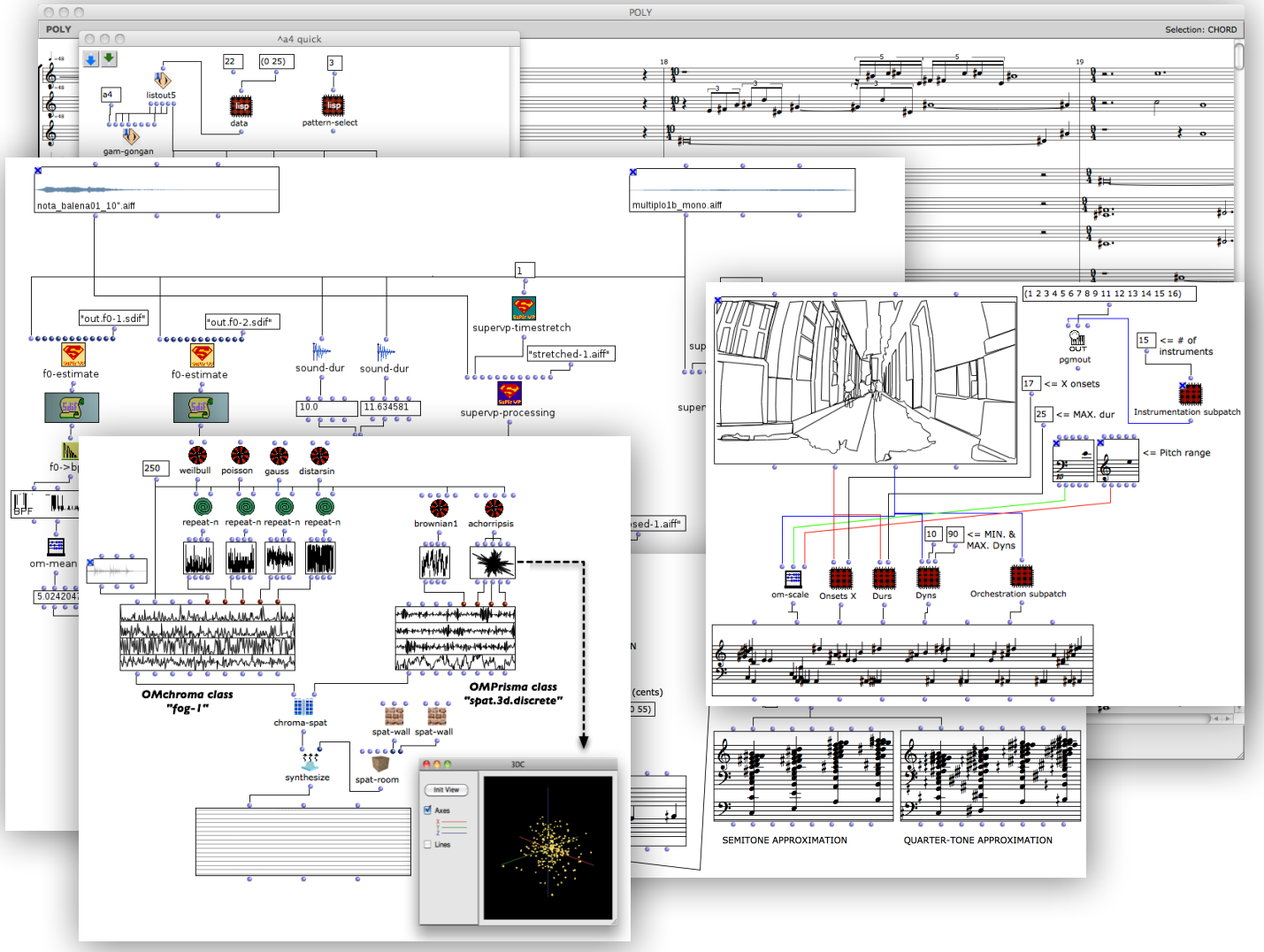
The OM Composer's Books

J. Bresson, C. Agon, G. Assayag (Eds.)
Collection Musique / Sciences
IRCAM - Editions Delatour France



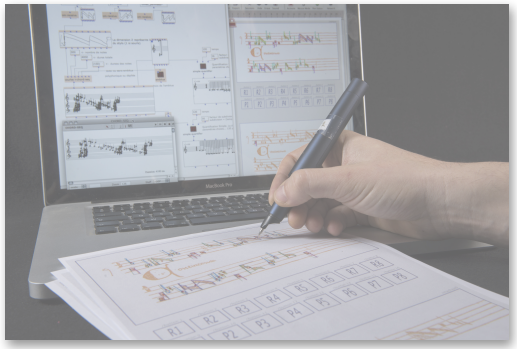


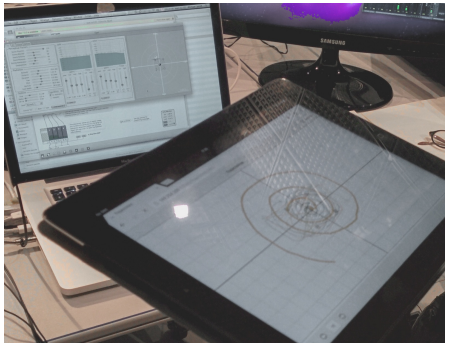
Volume 3: 2016 !

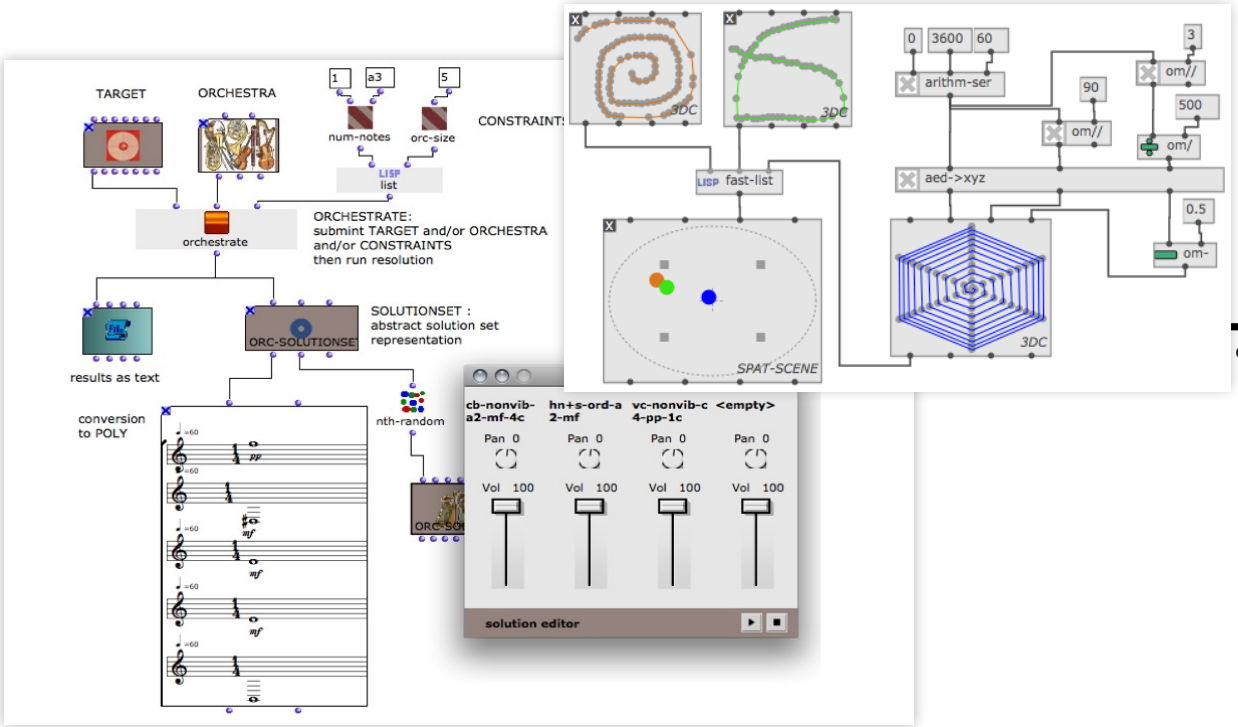


OSC communication with external systems in OpenMusic

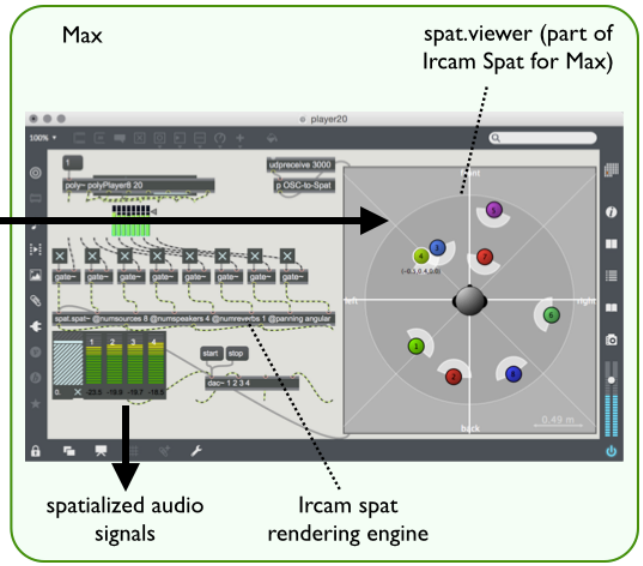
- **External input devices**
- **Control of spatial audio or real-time DSP**
- **Orchestration server**
- **Human-Computer Improvisation**
- **etc.**





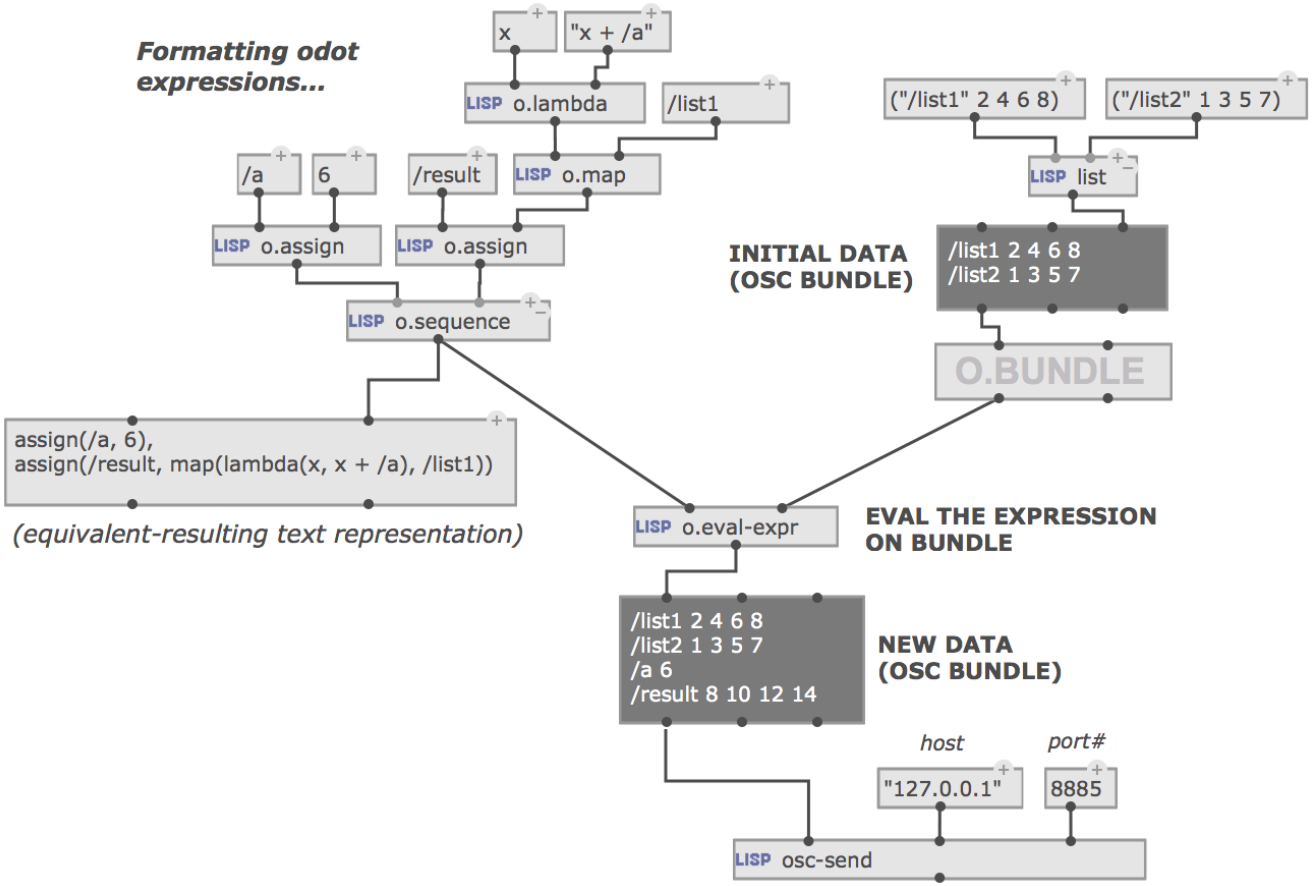


OSC control



- J. Garcia, J. Bresson, M. Schumacher, T. Carpentier, X. Favory: **Tools and Applications for Interactive-Algorithmic Control of Sound Spatialization in OpenMusic.** *InSonic*, Karlsruhe, 2015.

o.OM : An implementation of odot in OM

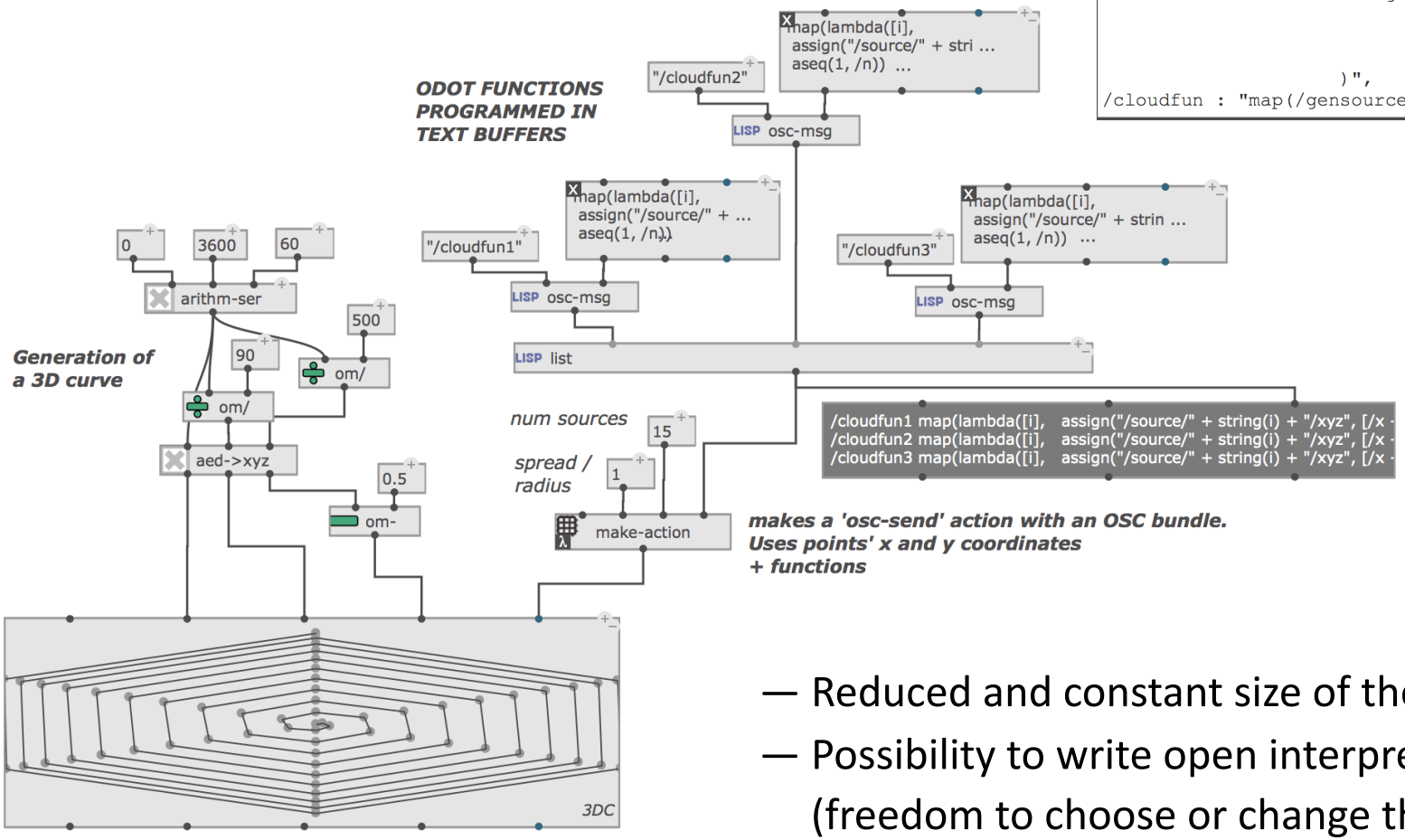


- **Create — Process — Send — Receive OSC messages and bundles**
- **Format expressions in the *odot* language**
- **Evaluate *odot* expressions on OSC bundles**

o.OM : examples of application

Use case 1: Control of realtime spatial audio synthesis

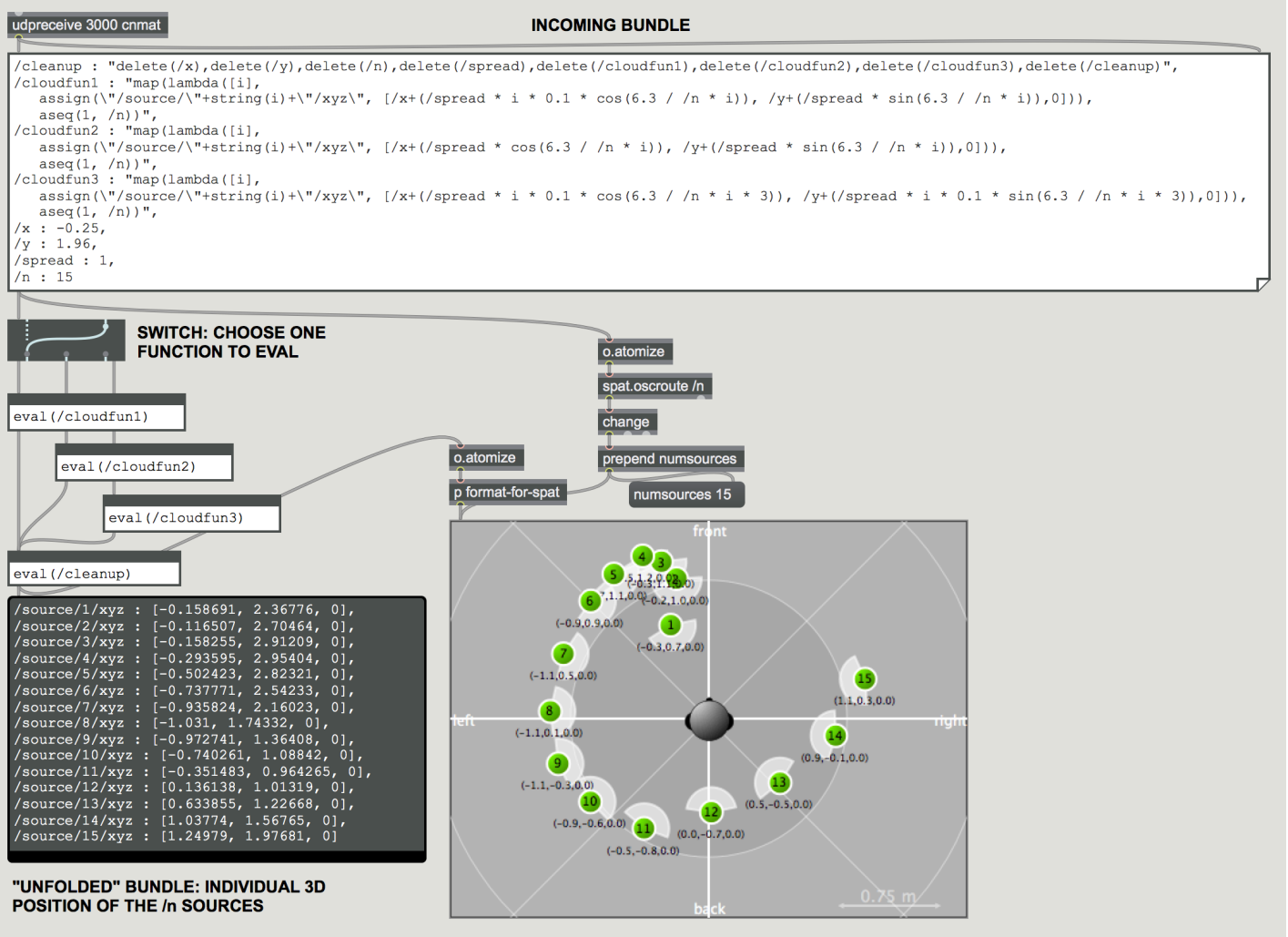
```
/x : -0.25,
/y : 1.96,
/spread : 1,
/n : 15,
/gensource : "lambda([i],
                     assign(\"/source/\"+string(i)+\"/xyz\",
                            [/x+(/spread*cos(/n*i)),
                             /y+(/spread*sin(/n*i)),
                             0])
                     )",
/cloudfun : "map(/gensource, aseq(1, /n))"
```



— Reduced and constant size of the streamed bundles

— Possibility to write open interpretive instructions, (freedom to choose or change the interpretation)

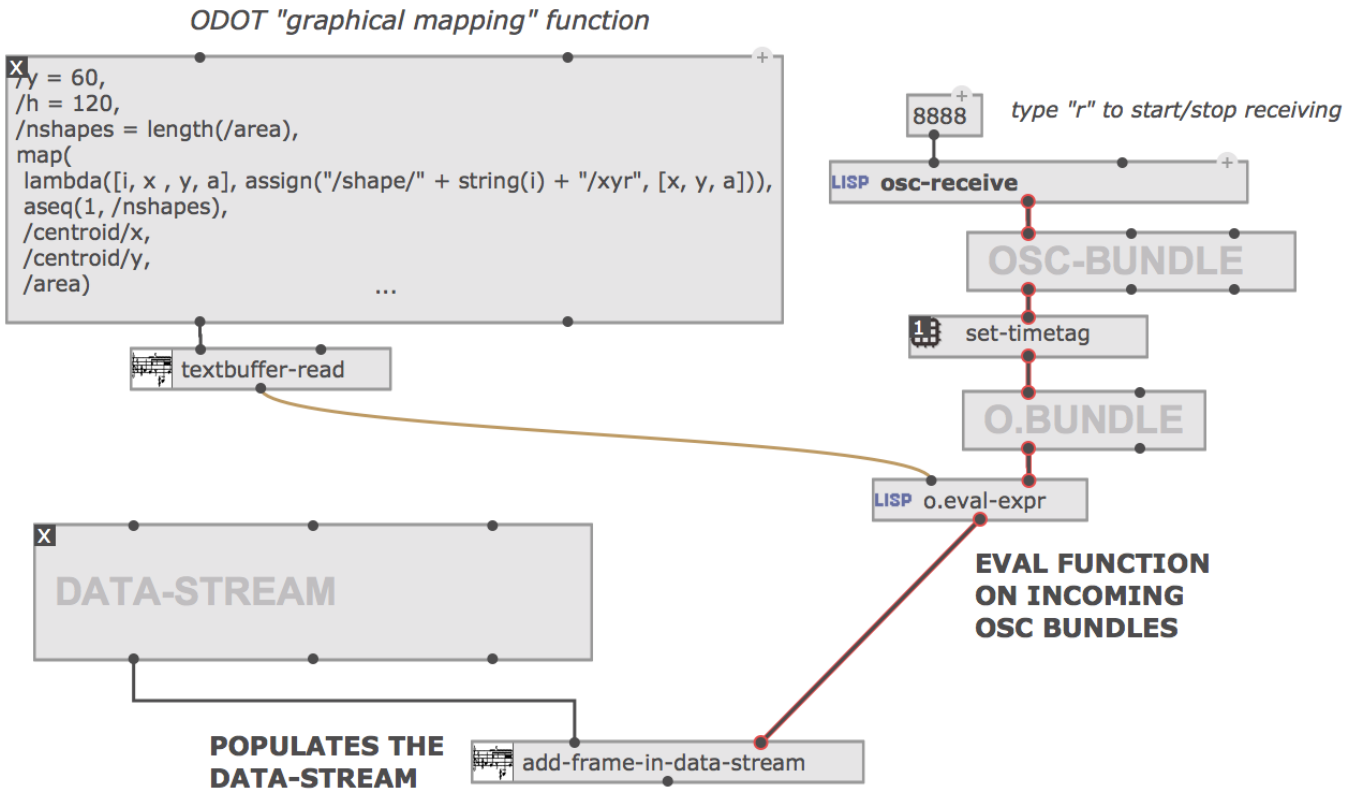
Use case 1: Control of realtime spatial audio synthesis



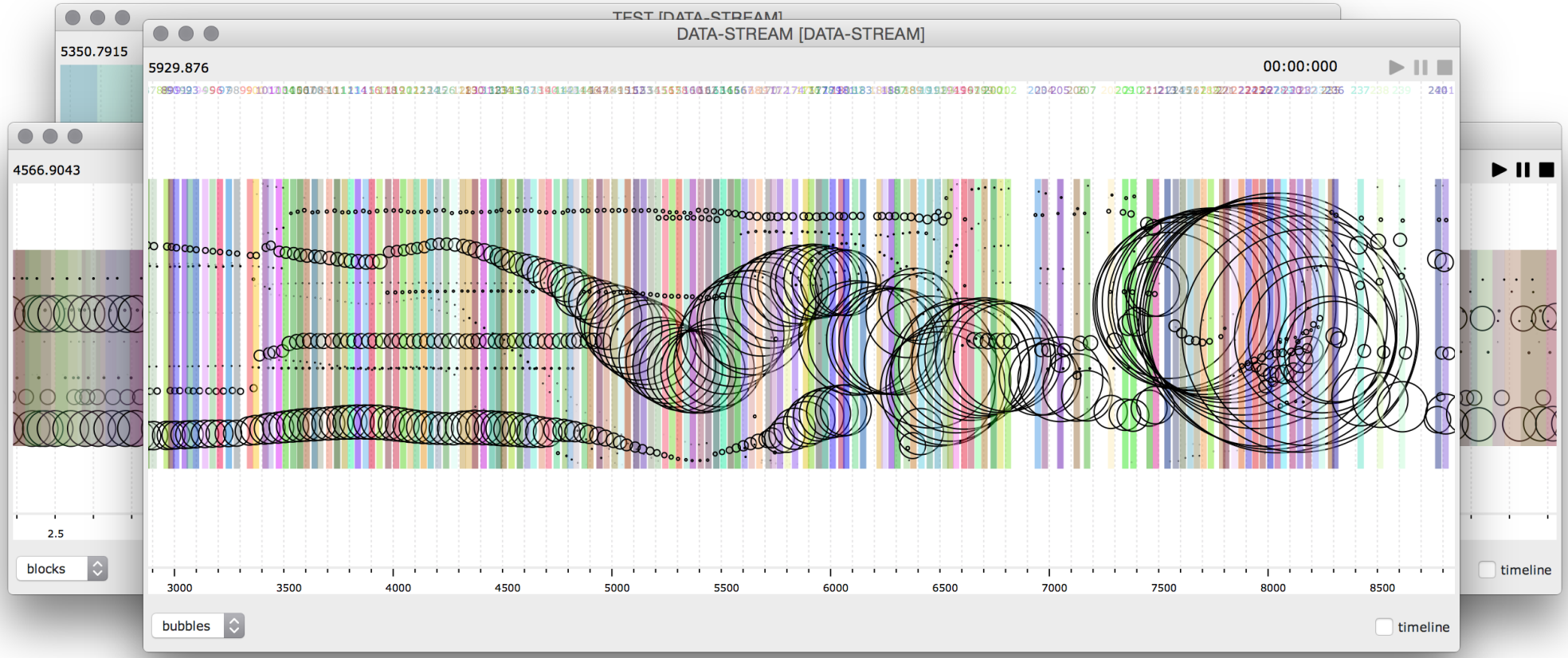
o.OM : examples of application

Use case 2: Receiving reception contour recognition data computed from live video capture

- Odot expressions used as mapping to graphical representation
- OSC bundles collected in a *data-stream* container



o.OM : examples of application



— Sequencer / timeline view of the OSC stream (visualisation and authoring)

— Possibility to personalise to representation (from OM or from outside, in the same language)

Advantages of using o. in OM/media environments:

— Embedding functional specification and programming within OSC

— Optimize and extends expressivity in communication frameworks

— Operate directly on the transferred data

— Sharing a common language between heterogenous environments

thank you :)