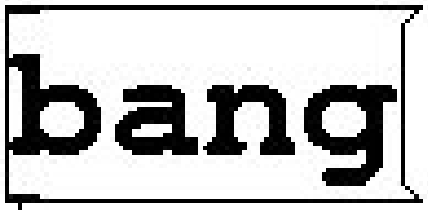
Pure Data workshop

Make art / Goto 10



Nicolas Montgermont
Cyrille Henry

2007 04 03

Présentation de la semaine

- Présentation
- La physique du son
- Les modèles physiques
- Les modèles physiques pour le mapping
- Les relations audiovisuelles

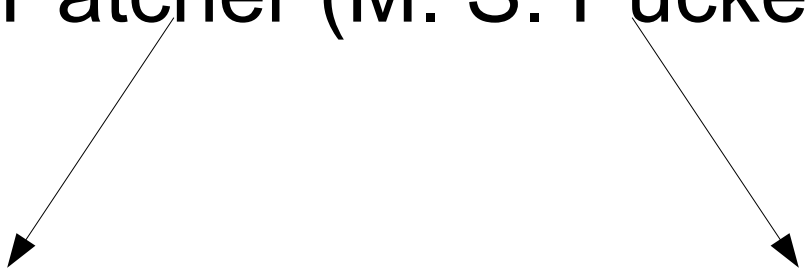
Exemples d'utilisation

- Modélisation
- Instrument augmenté
- Installation interactive
- Performance audiovisuelle
- Captation gestuelle

Introduction : Historique

Années 60-80 : Music I-V (Max Mathews)

1988 : Patcher (M. S. Puckette)



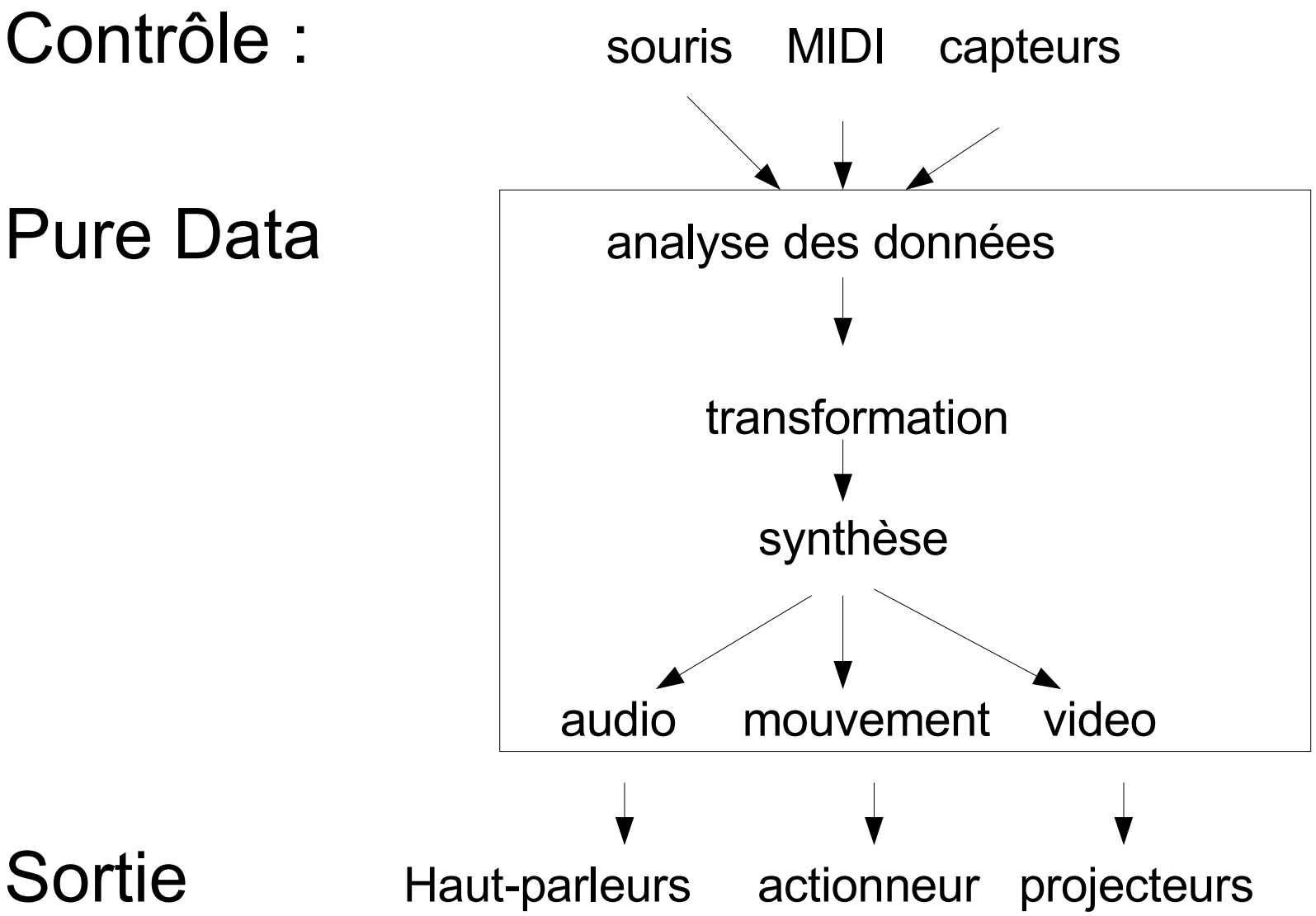
Max/ MSP

D. Zicarelli

Pure Data

M.S. Puckette

schéma général

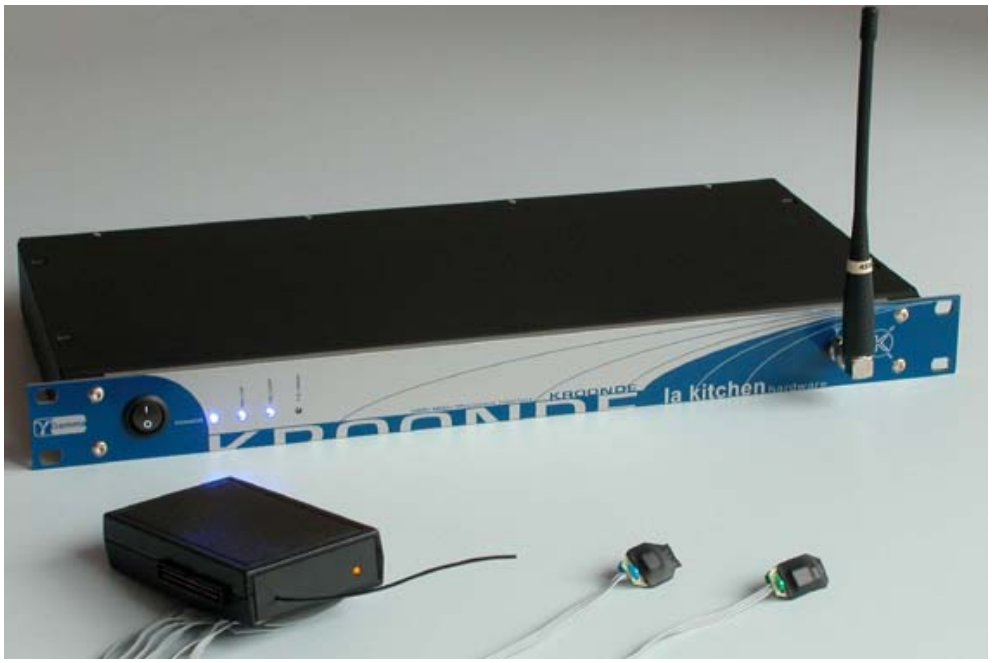


Temps Réel

- interactions en direct
- édition lors de l'exécution
- langage interprété
- cf processing, vvvv, usine, max/MSP/jitter, quartz composer, supercollider...
- notion théorique floue
- temps de rection du logiciel non perceptible

Capteurs / Actionneurs

- http://www.la-kitchen.fr/kitchenlab/kitchenlab.htn
- http://www.interface-z.com/
- http://www.bluemelon.org/index.php/Products/Bl

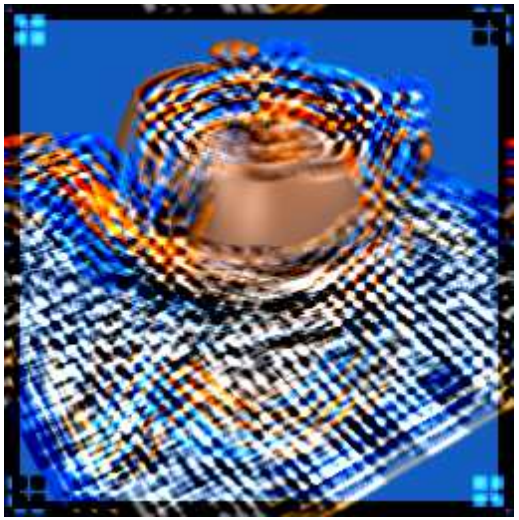
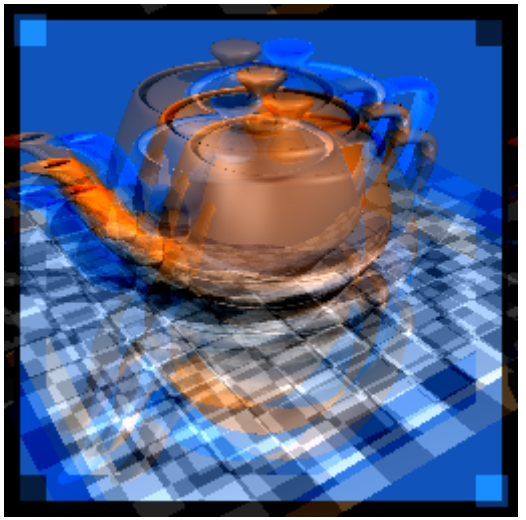
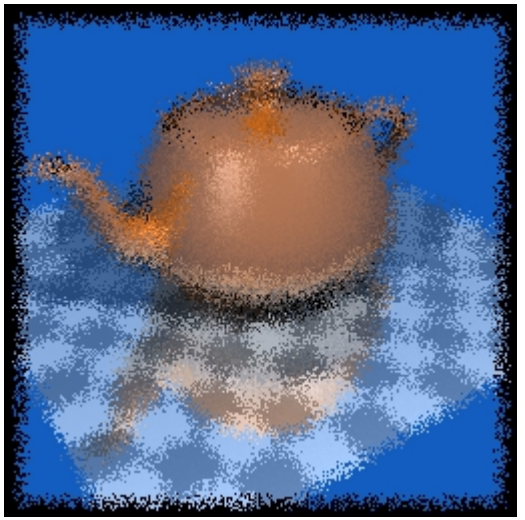


Synthèse Sonore

- http://puredata.info/
- http://www.crca.ucsd.edu/~msp/techniques.htm
- http://crca.ucsd.edu/~jsarlo/pdvst/

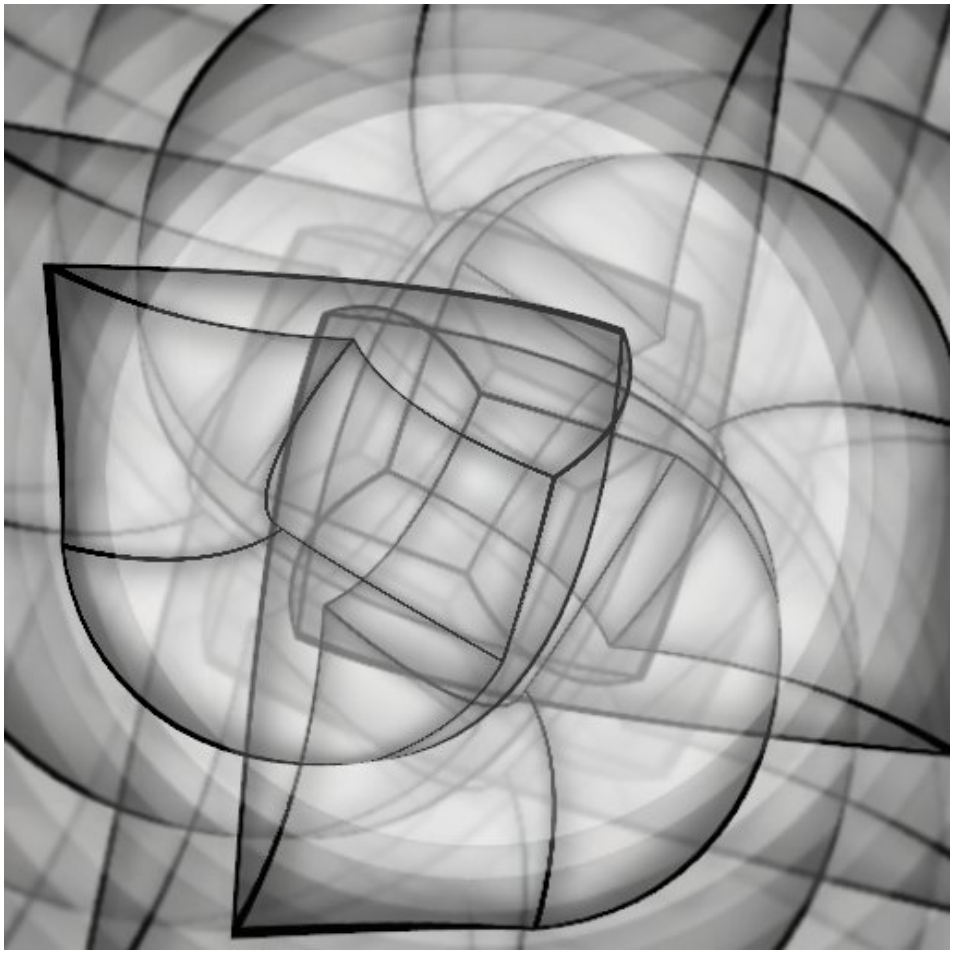
Traitement vidéo

- http://artengine.ca/gridflow/
- http://zwizwa.fartit.com/pd/pdp/

Synthèse visuelle

- http://gem.iem.at/



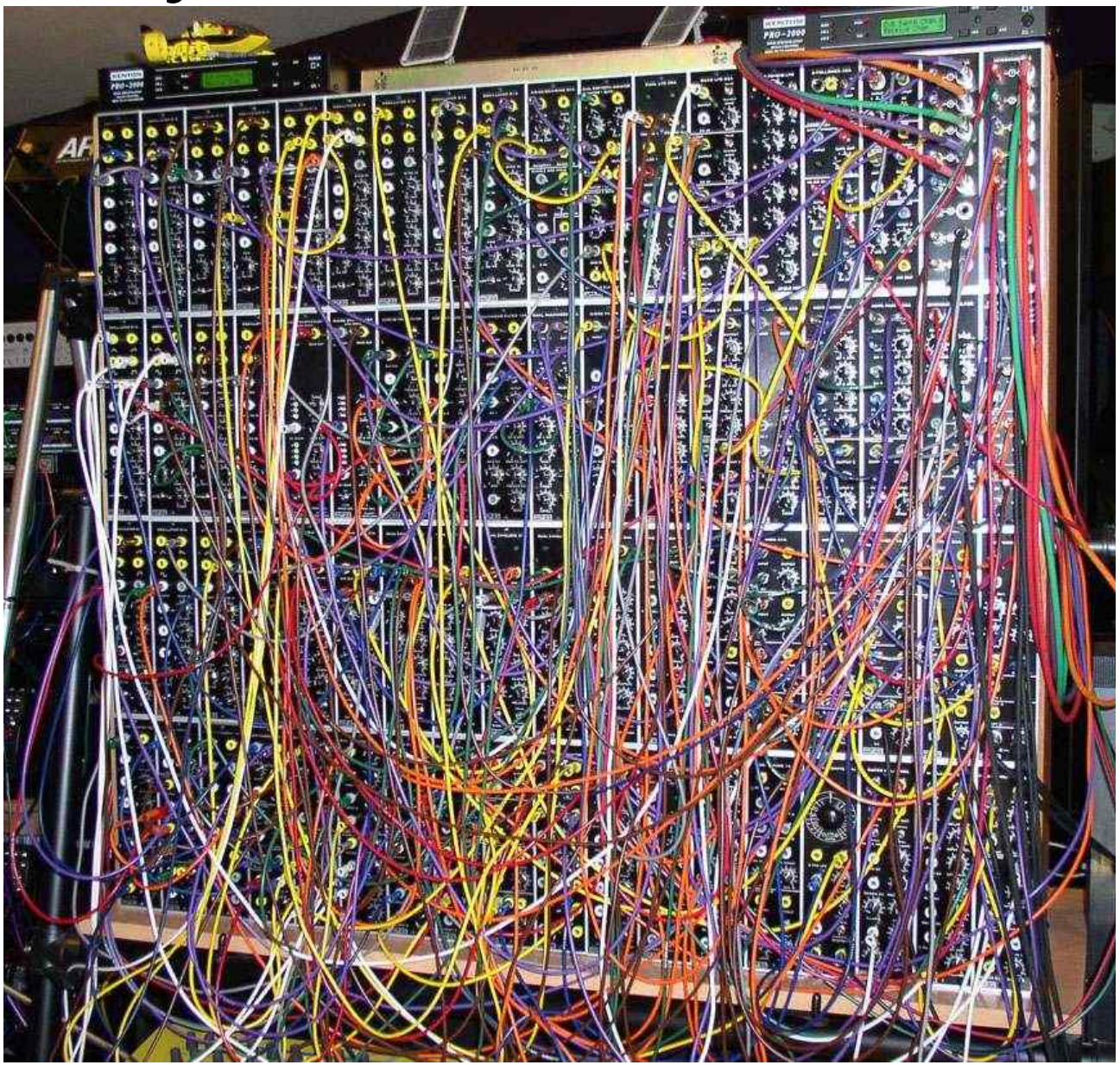
http://www.ekran.org/ben/

Système modulaire

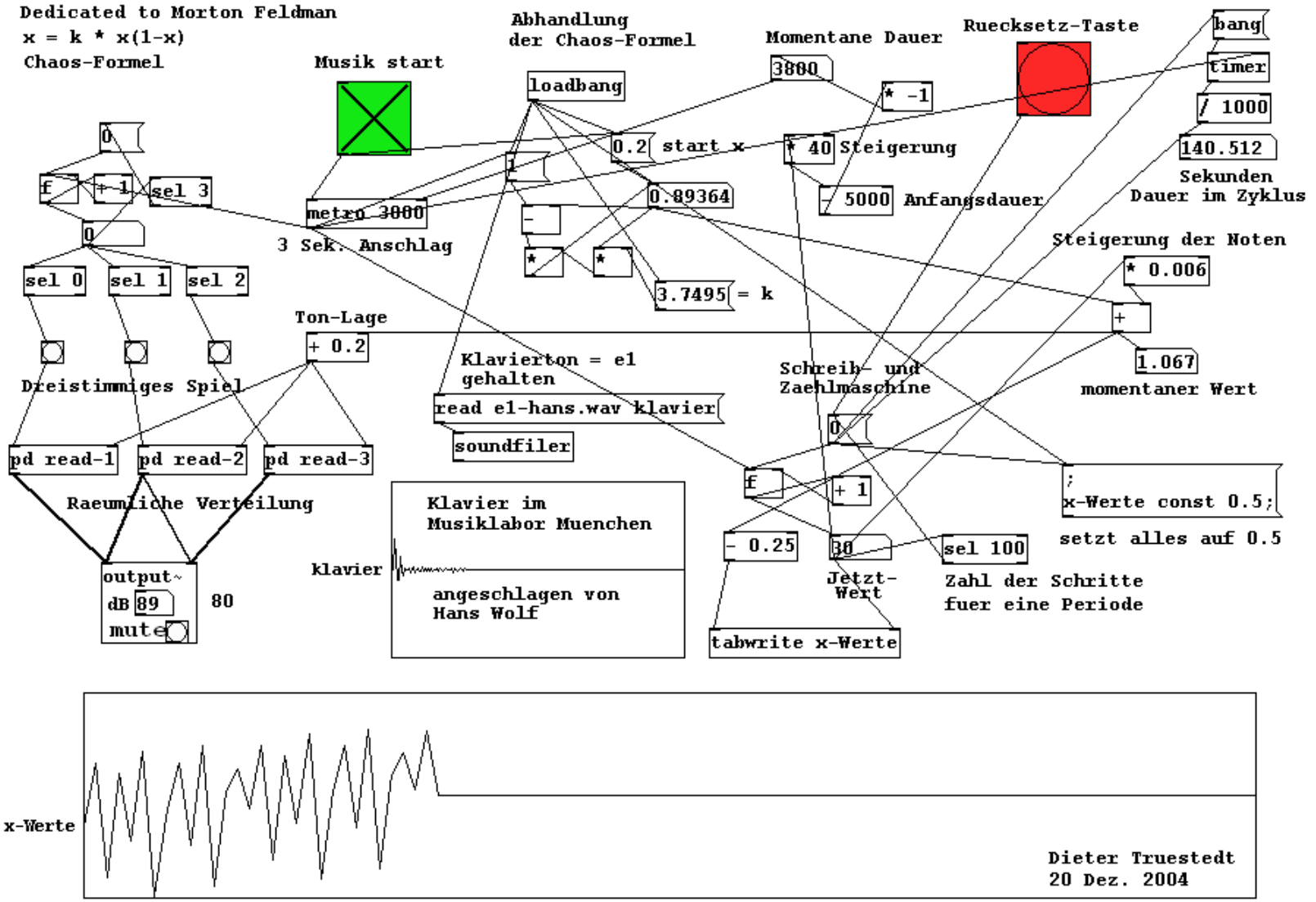
- Même principe que les synthétiseurs analogiques

Système Modulaire

Système Modulaire



Système Modulaire



Système ouvert

- Création de nouveaux modules
- Programmation en différents langages
- http://iem.kug.ac.at/pd/externals-HOWTO/
- http://www.le-son666.com/software/pdj/
- http://grrrr.org/ext/flext/

Logiciel libre

- licence BSD
- possibilité de modifier / distribuer les sources
- http://fr.wikipedia.org/wiki/Licence_libre
- http://m.ash.to/pd/
- https://puredyne.goto10.org/

Cross-platform

- Linux / Windows / osX / Irix / ...
- http://gige.xdv.org/pda/
- http://ipodlinux.org/PdPod

Ressources : en ligne

Téléchargement du logiciel :

Version de M. Puckette : http://www-crca.ucsd.edu/~msp
Version de Hans : http://at.or.at/hans/pd/installers
CVS : http://sourceforge.net/projects/pure-data

Communauté :

Officiel : http://puredata.info
mailing-list : http://lists.puredata.info/pipermail/pd-list
wikipedia : http://fr.wikipedia.org/wiki/Pure_Data
liste de liens : http://bill.teamtechno.com/pdspider/

Et maintenant, au travail....

Environnement de développement

Introduction

Exemples d'utilisation

Environnement de développement

- Le patch
- Le terminal
- Les externals

Paradigmes de programmation

Ressources

Exemples temps-réels

Environnement de développement : le patch

Le patch est une feuille de travail.

Il fonctionne selon deux modes: édition/ action

La programmation est graphique. Les fonctions sont réalisées en plaçant des boîtes, les variables circulent sur des cables

Les patchs sont utilisables dans d'autres patchs sous forme d'abstractions.

Environnement de développement : le terminal

- Fenêtre principale de Pd

- Correspond au stdout

- Gestion des erreurs

- Possibilité d'activer ou non le calcul audio

Environnement de développement : les externals

Les externals sont des objets codés dans un autre environnement.

Ils sont réalisés en C/ C++ et compilés pour Pd

Une grande communauté de développeurs existe et distribue ses externals : OSC, GEM, pmpd...

Paradigmes de programmation

Introduction

Exemples d'utilisation

Environnement de développement

Paradigmes de programmation

- Nature des boîtes
- Types de signaux
- Séquencage

Ressources

Exemples temps-réels

Paradigmes : Nature des boîtes

Les boîtes utilisables sont de 5 types :

- Atomes
- Objets
- Messages
- Interfaces graphiques
- ... le bang

Paradigmes : Nature des boites

Les **atomes** représentent les types de données de base.

Ils sont de trois types :

- nombre à virgule flottante.
- chaîne de caractère.
- liste.

Les **messages** transportent des chaînes de caractères

Paradigmes : Nature des boîtes

Les **objets** réalisent des fonctions.

Ils ont un nombre d'entrées et de sorties correspondant à la fonction à réaliser.

Les variables peuvent être transmises par des connections, ou initialisées à la création.

Ils sont de trois types:

- compilés.
- abstraction.
- sous-patch.

Paradigmes : Nature des boîtes

Les objets **graphiques** servent à interfacer le logiciel et l'utilisateur.

Ils peuvent servir de contrôle d'un paramètre ou de représentation de celui-ci.

Ils sont très coûteux en CPU.

Leurs propriétés graphiques sont éditables dans une faible mesure.

Paradigmes : les types de signaux

Les connections transportent des données de deux types:

- les signaux de contrôle. Ils sont transmis lors d'une demande spécifique.

- les signaux audios. Les connections audios transportent en permanence un flux dont les propriétés dépendent des réglages de la carte son (44.1 kHz, 16 bits). Les objets audios s'écrivent avec un tilde ~ et les connections sont en gras.

Paradigmes : séquencage

Le signal « **bang** » sert à déclencher les actions.

Les entrées des objets ne sont pas équivalentes:

- l'entrée de gauche est l'entrée **chaude**. Elle commande l'action.

- les entrées de droites dont les entrées **froides.** Elles servent à stocker des arguments.

Paradigmes : séquencage

Il peut y avoir indétermination sur l'ordre des actions à effectuer

Pour lever l'indétermination, on utilise l'objet **trigger** qui ordonne les événements de droite à gauche.