



HAL
open science

Actes de la 1^{ière}s journées du projet TIC

Laurent Capocchi, Jean François Santucci, Bastien Poggi

► **To cite this version:**

Laurent Capocchi, Jean François Santucci, Bastien Poggi. Actes de la 1^{ière}s journées du projet TIC. 1^{ière}s journées du projet TIC, Jun 2012, Corte, France. EARSeL eProceedings, 1 (1), 2012. hal-01083165

HAL Id: hal-01083165

<https://hal.science/hal-01083165>

Submitted on 15 Nov 2014

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.

Public Domain

Les journées du TIC

An abstract graphic featuring colorful, flowing waves in shades of green, yellow, and purple on the left, transitioning into black splatters and dots on a white background on the right.

1^{ières} journées du projet TIC

Proceedings 2011-2012

Ce document présente les actes des premières journées du Projet TIC qui se sont déroulées les 11 et 12 juin 2012 au sein de l'UFR FST de l'Université de Corse Pasquale Paoli.

Comité Editorial

L. Capocchi

J.F. Santucci

B. Poggi

Corte le 12 juillet 2012

“Favoriser la convergence des thématiques de recherche traitées dans le Projet TIC”

Les 1 ères Journées du Projet TIC ont eu lieu les 11 et 12 juin 2012 à l'Université de Corse. Le projet TIC a su depuis sa création en 2008 se développer jusqu'à atteindre une phase de maturité permettant la capitalisation des recherches autour des concepts de Modélisation et Simulation. Ces journées sont l'occasion pour le projet TIC de mettre en avant les travaux de ses membres et de favoriser des thématiques de recherche communes. Ces journées ont pour objectifs de susciter des échanges autour des travaux de chacun des membres et de valoriser ces travaux dans le cadre de collaborations. Le concept de Brainstorming et les développements de travaux scientifiques sous la forme d'ateliers ont offert un cadre propice aux discussions, lors desquelles des concepts et même des logiciels innovants ont émergé. C'est l'occasion pour tous ses membres d'explorer les thématiques du projet TIC, et d'éventuellement initier des collaborations communes. Cette année, ce ne sont pas moins de 17 participants qui ont couverts l'ensemble des thématiques du projet TIC.

Ces journées n'existeraient pas sans la détermination et la disponibilité des membres du projet TIC. Leur motivation fût essentielle à la concrétisation de cette première édition. Qu'ils en soient toutes et tous remerciés; leur participation est un signe de leur implication au sein du projet. Nous vous donnons rendez-vous en 2013 pour la seconde édition.

Le responsable du projet TIC.

Jean-François Santucci



Les journées du TIC



Sommaire

<u>1.Programme.....</u>	<u>4</u>
<u>2.Ateliers.....</u>	<u>5</u>
<u>2.1.Atelier 1 : Ingénierie des modèles.....</u>	<u>5</u>
<u>2.2.Atelier 2 : Modélisation comportementale d'une population d'oursins dans un milieu donné à l'aide du formalisme DEVS.....</u>	<u>6</u>
<u>2.3.Atelier 3 : Bibliothèque DEVSimPy pour la simulation de données floues</u>	<u>7</u>
<u>3.Diapositives.....</u>	<u>8</u>
<u>3.1.Atelier 1.....</u>	<u>8</u>
<u>3.2.Atelier 1 (suite).....</u>	<u>9</u>
<u>3.3.Atelier 2.....</u>	<u>10</u>
<u>3.4.Atelier 3.....</u>	<u>11</u>

1. Programme

Le 11 Juin 2012

9h30-10h : Accueil

10h-12h 1ere partie : Chaque membre présente pendant 5 mn une proposition d'atelier

12h-13h30 : Dejeuner

13h30-15h30 2eme Partie: Brainstorming pour définir les ateliers

15h30-18h00 3eme Partie: Ateliers Développement

Le 12 juin 2012

9h-12h : 3eme Partie (suite et fin) : Ateliers Développement

12h-13h30 : Dejeuner

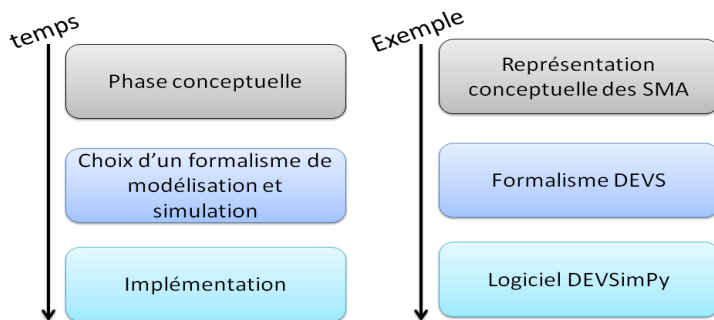
13h30-15h 4 ième partie: Préparation de la restitution des resultats par atelier.

15h-17h : 5ième partie : Restitution des travaux sous différente formes – code, plan publis,etc...,

2. Ateliers

2.1. Atelier 1 : Ingénierie des modèles

Participants : E. Vittori, E. Broutin, M. Delhom, S. Mattei, **B. Poggi**, S. Garredu

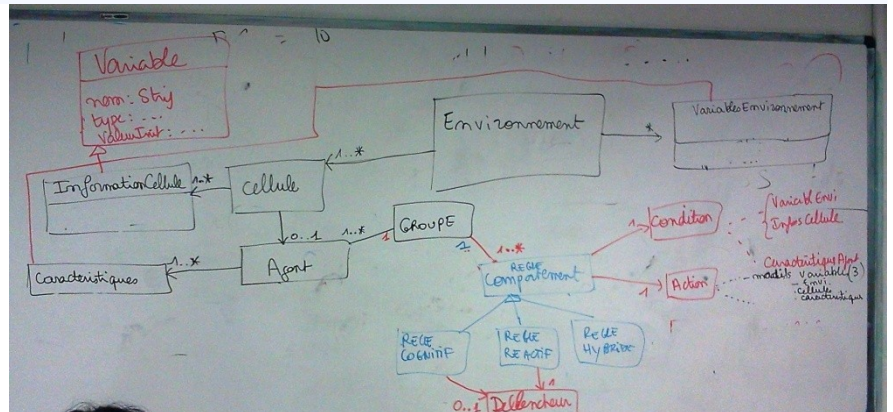


L'atelier ingénierie des modèles s'est déroulé autour de la création d'un Métamodèle DEVS permettant la modélisation de systèmes multi-agents avec comme perspective l'implémentation de la dynamique des populations.

Le Métamodèle permet une description générique d'un système. Plusieurs niveaux de méta-modélisation sont possibles comme visible sur la figure ci-dessus. Le Métamodèle est indépendant de tout formalisme et se doit d'être compatible avec chacun d'eux. Cette représentation permet d'assurer un haut niveau de qualité durant tout le processus de modélisation.

Dans un premier temps nous avons isolé les différents composants d'un système multi-agent comme vous pouvez le voir sur la figure de droite.

- Agent et groupe
- Règle comportementale
- Environnement
- Cellule
- Informations agents
- Informations cellule
- Actions et architectures



Par la suite les différents membres de l'atelier ont été initié aux différents standards MDA (Model Driven Architecture) de l'ingénierie des modèles :

- Formalisme de méta-modélisation
- Représentation concrète de modèles
- Modélisation des transformations

Nos échanges ont également concernés l'aspect MDE (Model Driven Engineering) et les outils associés. Dans la continuité nous avons utilisé le framework EMF (Eclipse modeling framework). Celui-ci permet une définition aisée du Métamodèle à partir d'une interface graphique riche et intuitive. Il permet également à partir de spécifications (XML, EMF) l'auto-génération du code source des modèles.

Une fois le Métamodèle terminé, nous avons procédé à sa validation par la création d'un jeu de test. Nous avons ainsi pu vérifier la cohérence des modèles et de leur interactions. Ces travaux ont été sauvegardés dans un fichier Ecore pour pouvoir être réutilisés dans des travaux futures.

2.2. Atelier 2 : Modélisation comportementale d'une population d'oursins dans un milieu donné à l'aide du formalisme DEVS

Participants : A. Aiello, **L. Capocchi**, S. Toma, T.A. Santoni

L'objectif de cet atelier était d'initier une réflexion afin d'apporter une solution à la problématique d'observation et de prédiction du comportement d'une population d'oursins sur une zone donnée. Cette problématique est complexe car elle dépend de nombreux paramètres fournis par les spécialistes du domaine: les biologistes.

A l'heure actuelle, les biologistes possèdent un ensemble de données d'observation et de mesures sur l'espèce permettant d'affirmer les informations suivantes:

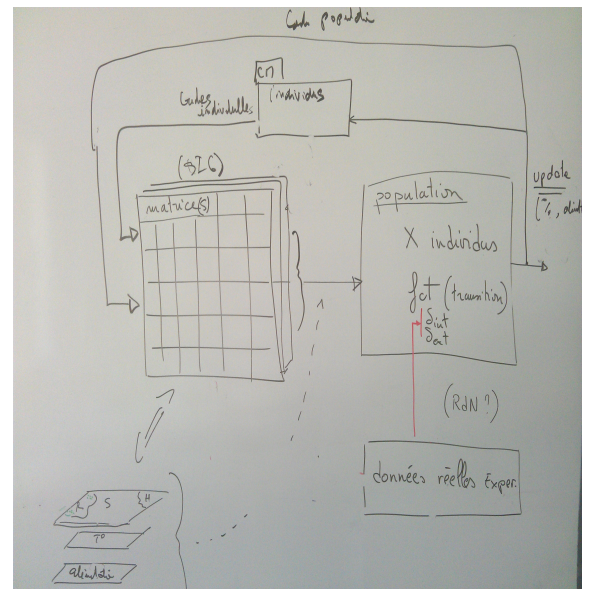
- Profondeur max d'évolution : 9 mètres
- Vitesse de déplacement : 0 à 260 cm/j
- Moyenne de déplacement : 50 cm/j
- Orientation de la vitesse en fonction du milieu ($V_r > V_h$) et turbulences (courant)

L'approche devra utiliser le formalisme DEVS pour son cadre formel et son application dans l'environnement DEVSimPy.

Nous avons abordé deux types de modélisation possible: par individus (coordonnées x,y,z) ou par population.

Si l'acquisition de données permet de fournir la position des individus observés alors nous pouvons effectuer une transformation en donnée de zoning. Le choix retenu a été le modèle centré population du fait du manque de données à disposition immédiate.

L'orientation des travaux vers un modèle DEVS permettra de considérer le comportement d'une population d'individus (deterministe, réseaux de neurones,...) et de caractériser une zone observée par une approche matricielle (Cell-DEVS).



Les **perspectives** de réflexion sont les suivantes:

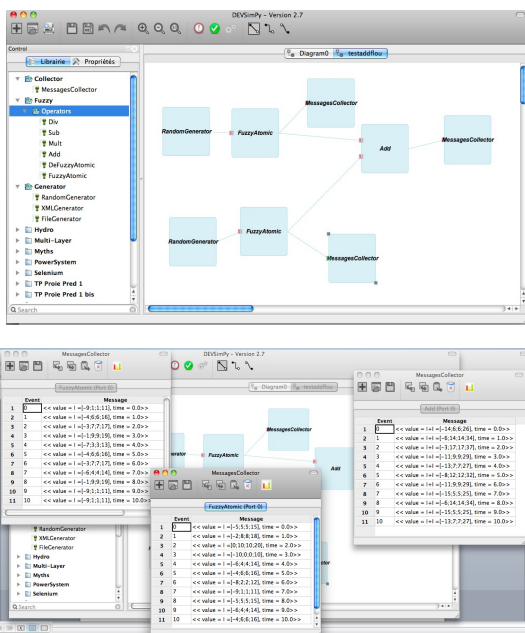
- Etape 1 :
 - * Mise en œuvre de l'acquisition de données.
 - * Déploiement sur un secteur test (200m2).
 - * Passage à l'échelle de terrain (4Km2).
- Etape 2 :
 - * Organisation de séminaire Projets Eau/TIC à Stella-Mare.
 - * Développement du modèle présenté à partir des spécifications des biologistes. et déploiement du simulateur.
- Etape 3 :
 - * Validation à partir des données expérimentales réelles.
 - * Amélioration du modèle.



2.3. Atelier 3 : Bibliothèque DEVSimPy pour la simulation de données floues

Participants : P.A Bisgambiglia, A.T. Luciani, P.A. Bisgambiglia, T. Ville, **J.F. Santucci**

Objectif : le but de l'atelier était de réfléchir à la définition d'une bibliothèque de modèles DEVSimPy permettant à un modélisateur de facilement traiter des données floues au sein d'un modèle DEVS.



Travail réalisé : nous avons donc tout d'abord défini au cours de l'atelier quels étaient les modèles à offrir à un modélisateur pour faciliter le traitement de données floues. Afin d'aider un modélisateur pour l'intégration de données floues au sein d'un modèle DEVS, il faut tout d'abord définir deux modèles atomiques permettant : (i) de transformer des données exactes en données floues – composant FuzzyAtomic ; (ii) de transformer des données floues en données exactes – composant DeFuzzyAtomic. Ensuite, il faut offrir dans la bibliothèque un ensemble d'opérateurs de base permettant le traitement de données floues : (1) l'addition – composant Add ; (2) la soustraction – composant Sub ; (3) la multiplication – composant Mult ; (4) la division – composant Div.

Nous avons donc implémenté ces six composants DEVS en langage Python au sein de l'environnement DEVSimPy.

Nous avons ensuite énuméré l'ensemble des autres composants à développer afin d'augmenter le nombre de modèles à offrir dans la bibliothèque.

Résultats : les six composants développés forment la bibliothèque Fuzzy de DEVSimPy comme nous pouvons le voir sur la partie gauche de la première figure ci-dessus. Nous avons ensuite validé ces modèles en construisant un modèle DEVS impliquant des données floues (partie droite de la figure ci-dessus). Ce modèle possède 2 générateurs de nombres entiers. Les sorties de ces générateurs sont fuzzyfiées grâce à deux modèles FuzzyAtomic. Les sorties (qui sont des données floues) de ces deux modèles sont alors additionnées – en effectuant bien sur une addition floue – grâce au modèle Add. Afin de visualiser si l'opération s'effectue comme il faut nous avons ajouté 3 modèles MessageCollector qui permettent de visualiser : (1) les données floues en entrée du composant Add ; (2) le résultat de la somme. La seconde figure ci-dessus présente les valeurs obtenues après simulation. Nous pouvons alors vérifier que tout est correct.

Perspectives : nous devons compléter la bibliothèque avec notamment un ensemble d'opérateurs permettant la comparaison entre données floues. Nous avons aussi évoqué la possibilité pour le modélisateur d'écrire le code DEVSimPy d'un modèle devant prendre en compte les données floues en utilisant un surcharge des opérateurs de base. Enfin il nous a semblé intéressant de proposer une transformation automatique d'un modèle traitant les données floues en interconnexion d'opérateurs de base issus de la bibliothèque.

3. Diapositives

3.1. Atelier 1

Atelier Ingénierie des Modèles
Journées Projet TIC
(11-12 juin 2012)
Stéphane Garreau - Evelyne Vittori
Eclipse EMF

Présentation de l'atelier

- Objectifs
- Qu'est ce que l'IDM?
- Quels apports
 - pour la Modélisation et Simulation?
 - dans le contexte DEVS?
- Une approche concrète de l'IDM :
 - l' Environnement de développement Eclipse Modeling FrameWork (EMF)

Objectifs de l'atelier

- Introduire les potentialités de l'ingénierie des modèles
- Faire découvrir l'environnement Eclipse EMF
- Réfléchir ensemble à des applications possibles dans le contexte du projet TIC

Qu'est ce que l'IDM?

- Un ensemble de technologies de génie logiciel favorisant la productivité et l'interopérabilité du développement
- Grands principes :
 - « Model once, generate everywhere »
 - « Tout est modèle »

Concrètement

- Définition de métamodèles
- Définition de transformations automatiques de modèles

Qu'est ce que l'IDM?

- Des standards : MDA (Model Driven Architecture)
- Formalisme de métamodélisation
- Représentation concrète de modélisation
- Modélisation des transformations

- Des outils, des environnements MDE (Model Driven Engineering)

Présentation de l'atelier

- Objectifs
- Qu'est ce que l'IDM?
- Quels apports
 - pour la Modélisation et Simulation?
 - dans le contexte DEVS?
- Une approche concrète de l'IDM :
 - l' Environnement de développement Eclipse Modeling FrameWork (EMF)

Quels apports pour la Modélisation et Simulation?

- Assurer l'interopérabilité à 2 niveaux :
 - Simulateurs
 - Formalismes de modélisation
- Faciliter le développement d'outils de définition de modèles
- Promouvoir la standardisation
 - Des formats de stockage des modèles
- Automatiser les transformations entre modèles exprimés en différents formalismes

Quels apports dans le contexte DEVS?

- Le métamodèle DEVS (publication en cours) :
 - Permet la standardisation des modèles
 - Construit le pivot de l'interopérabilité
- Elargissement : extensions de DEVS, voire métamodèles de formalismes d'interface
- Transformation d'un formalisme d'interface vers DEVS (= M2M +)
- Génération automatique de code (= M2T +)

Quels apports dans le contexte DEVS?

Diagram illustrating the relationship between DEVS, MDE, and modeling tools. It shows a central 'Métamodèle DEVS' box connected to 'MDE (Model Driven Engineering)' and 'Modélisation'. The diagram also shows 'Eclipse EMF' and 'Eclipse Modeling Framework' as part of the environment.

Présentation de l'atelier

- Objectifs
- Qu'est ce que l'IDM?
- Quels apports
 - pour la Modélisation et Simulation?
 - dans le contexte DEVS?
- Une approche concrète de l'IDM :
 - l' Environnement de développement Eclipse Modeling FrameWork (EMF)

Environnement Eclipse EMF

- Communauté OpenSource
 - Projet collaboratif très actif
 - Principe de plug-in
- EMF: Eclipse Modeling Framework
 - Ensemble d'outils dédiés à l'IDM
- Propose des approches concrètes IDM
 - Problématique de la définition d'un nouveau formalisme

Comment créer un nouveau formalisme avec EMF?

- Définir un métamodèle (grâce au méta-métamodèle Ecore)
- Différentes vues de la même structure qui se complètent :
 - Editeur Ecore arborescent
 - Editeur graphique « Ecore Diagram » DSL
 - Editeur OCLinEcore Domain Specific Language

Comment définir des modèles conformes à ce formalisme?

- Un modèle est une instance d'un métamodèle
 - format de stockage: fichier XML
- Comment créer des modèles?
 - Pendant la définition du métamodèle
 - Création dynamique d'instances (permet des tests)
 - Générer un éditeur arborescent
 - Création d'un plugin spécifique
 - Définir un éditeur graphique à l'aide de GMF (Graphical modeling Framework)
 - Création d'un plugin associé

Comment transformer les modèles?

- Transformation M2M (model to model)
 - Langages de définition des transformations
 - Transformations définies au niveau des méta modèles
- Standard OMG: QVT (Query/View/Transformation)
 - ATL: Atlas Transformation Language
 - Implémentation de QVT
 - Plugin Eclipse associé
 - Langage de règles

Mise en œuvre de l'interopérabilité

Comment générer du code?

- Transformation M2T: model to Text
- Standard OMG : MOF Models to Text Transformation Language (MOFM2T)
 - Implémenté par le plugin ACCELEO : utilise le principe des templates

Difficultés du domaine

- Problématique du choix des outils et de leur mise en synergie
- Complexité :
 - investissement important nécessaire pour maîtriser les concepts et les outils
 - Bénéfices perceptibles à moyen et long terme uniquement

Approches proposées dans l'atelier

- Etudier le Métamodèle DEVS Ecore
- Réfléchir à des formalismes d'interface
 - Métamodèles
 - Transformations M2M vers DEVS ATL
- Travailler sur la génération de code vers DEVSIMPY Acceleo

3.2. Atelier 1 (suite)

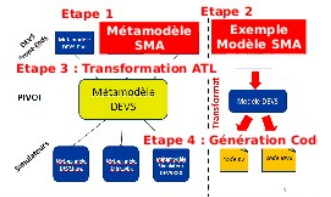
Objectifs de l'atelier

- Mise en œuvre d'une démarche d'ingénierie des modèles dans le contexte DEVS
- Application aux SMA en tant que formalisme d'interface destiné à être transformé vers DEVS
- Implémentation dans l'environnement Eclipse EMF

Démarche de travail

1. Définition et implémentation d'un métamodèle SMA
2. Validation par définition de modèles de tests
3. Définition de règles de transformation vers DEVS
 - Programme ATL
4. Génération de code dans DEVSIMPY

Démarche de travail



Ce que nous avons réalisé

1. Réflexion sur les principaux concepts des SMA
2. Définition d'un métamodèle associé
 - Prise en compte des agents réactifs
 - Métamodèle extensible
3. Implémentation dans l'environnement Eclipse EMF
 - Richier Ecole
5. Définition de modèles de tests (instances du métamodèle)

Ce qu'il nous reste à faire

- ▶ Validation du métamodèle sur des exemples complexes
- ▶ Enrichissement du métamodèle SMA
 - ▶ Agents cognitifs et hybrides
 - ▶ Architectures avancées (BDI, ...)
- ▶ Réalisation de la transformation en ATL
- ▶ Différentes approches envisageables

Perspectives et valorisation

- ▶ Rédaction d'un article
- ▶ Possibilité de généralisation de la démarche à d'autres formalismes (ex: réseaux de Petri)

3.3. Atelier 2

Journées du TIC

Atelier : Modélisation comportementale d'une population d'oursins dans un milieu donné à l'aide du formalisme DEVS

A. Aiello, T.A. Santoni, L. Capocchi, S. Toma

Journées du TIC – 11-12/06/2012

Reflexion

- Deux orientations de modélisation possibles :
 - par individu (position x,y,z).
 - par population.
- Si l'acquisition de données permet de fournir la position des individus observés → transformation en donnée de zoning.
- Choix retenu :
 - Modèle centré population du fait du manque de données à disposition.

Journées du TIC – 11-12/06/2012

Perspectives

- **Etape 1**
 - Mise en œuvre de l'acquisition de données.
 - Déploiement sur un secteur test (200m²).
 - Passage à l'échelle de terrain (4Km²).
- **Etape 2**
 - Organisation de séminaire Projets Eau/TIC à Stella-Mare.
 - Développement du modèle présenté à partir des spécifications des biologistes.
 - Déploiement du simulateur.
- **Etape 3**
 - Validation à partir des données expérimentales réelles.
 - Amélioration du modèle.

Journées du TIC – 11-12/06/2012

Objectif

- Observer et prévoir le comportement d'une population d'oursins sur une zone donnée :
 - En fonction de paramètres complexes issus des spécialistes du domaine : biologistes.
 - En utilisant le formalisme DEVS couplé à un système d'acquisition de données en temps réel.

Journées du TIC – 11-12/06/2012

Proposition

- Orientation des travaux vers un modèle DEVS permettant :
 - De prendre en compte le comportement d'une population d'individus (Déterministe, RN, ...).
 - De caractériser une zone observée par une approche matricielle (Cell-DEVS).

Journées du TIC – 11-12/06/2012

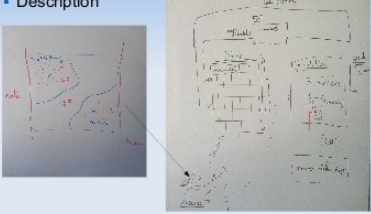
Constats

- Le phénomène observé est fonction de données biologiques complexes qui seront à préciser (cf. publications en attente S. Ternengo et V. Pasqualini)
- Premiers éléments biotiques et abiotiques :
 - Profondeur max : -9 mètres.
 - Vitesse de déplacement : 0 à 260 cm/j.
 - Moyenne de déplacement : 50 cm/j.
 - Orientation vitesse en fonction du milieu ($V_r > V_h$) et des turbulences (courant).

Journées du TIC – 11-12/06/2012

Proposition (suite)

- Description



Journées du TIC – 11-12/06/2012

3.4. Atelier 3

Bibliothèque
DEVsimPY
Logique Floue

Paul-Antoine Bigambiglia – Jean-François Santucci
Paul-Antoine Bigambiglia – André Toussaint Luciani
Timothée Yvinec

Introduction

- Prendre en compte la simulation de données floues
- Développement de modèles atomiques facilitant la réalisation de modèles plus complexe par Drag&Drop
- Traitement flou de données => Traitement flou sur le temps
- Exemple pédagogique

To_Disk

Application

Modèles

- DeFuzzy
- Fuzzy
- Opérateurs
- To_Disk flou

DeFuzzy

Application

Conclusion

- Nouvelle librairie dans DEVsimPY
- Projet d'article

Fuzzy

Opérateurs

Perspectives 1/3

- Modèles :
 - To_Disk flou
 - Min | Max | Comparateur
 - Opérateurs flous complexes (cos, sin, tan...)
 - SupMin

Perspectives 2/3

Perspectives 3/3

- Etude de la transformation du flou sur les données en flou sur le temps
- Exemple : Modèle Sol
- En théorie : la fonction ta est déclenchée suivant une fonction floue obtenue à partir du flou sur les données. La règle de transformation permet de générer un ta flou
- L'imprécision sur une valeur peut impliquer une imprécision sur le ta (-1,0,1)