

Tache d'assemblage de modèles 3D de montres

```
$ git clone git@github.com:rosconfr24-ws4/acroba.git  
$ cd acroba
```

Ou pour la mise à jour du setup existant:

```
$ cd acroba  
$ git pull
```

Dans les deux cas:

```
$ make pull  
$ make setup
```

Workshop 4

Création d'une tâche d'assemblage à l'aide d'arbres de comportement

Laurent Cavazzana



Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

1. Le projet ACROBA

2. Les Arbres de Comportement

3. Tache d'assemblage de modèles 3D de montres

Ère de l'Industrie 4.0

- Évolution rapide des marchés
- Hyper-personnalisation des produits

Ère de l'Industrie 4.0

- Évolution rapide des marchés
 - Hyper-personnalisation des produits
- ➔ Défis pour l'industrie manufacturière

Ère de l'Industrie 4.0

- Évolution rapide des marchés
- Hyper-personnalisation des produits
- ➔ Défis pour l'industrie manufacturière
- réduire le délai de mise sur le marché
- reprogrammation des outils de production ?

Fabrication Agile

- concept de production flexible
- Système robotique capable de s'adapter
- Reprogrammation facilité
- Collaboration homme–robot (HRC)
- Clone digital

Le projet ACROBA

- + créer une plateforme robotique cognitive
- + facilement configurable
- + adaptable à pratiquement tous les scénarios industriels

Le projet ACROBA

- + créer une plateforme robotique cognitive
 - + facilement configurable
 - + adaptable à pratiquement tous les scénarios industriels
- ➔ Réduction du temps de configuration, de programmation et de mise en service d'une solution robotique

Le projet ACROBA

**H2020
Innovation action**

**~8M€ budget
~7M€ EU funding**

5 industrial pilots

**Coordinator:
BFH**

42 months

**2 ACROBA
On-Site Lab**

**17 partners
9 countries**

Reference
Architecture
COPRA-AP

12 hackathons

Le projet ACROBA

1 large company



7 SMEs



5 research centers



2 clusters



2 universities



The ACROBA project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017284.

ROSCON FR Nantes – Workshop 4

Création d'une tâche d'assemblage à l'aide d'arbre de comportement

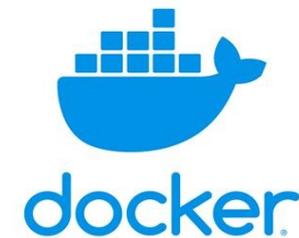


ROSCON FR Nantes – Workshop 4

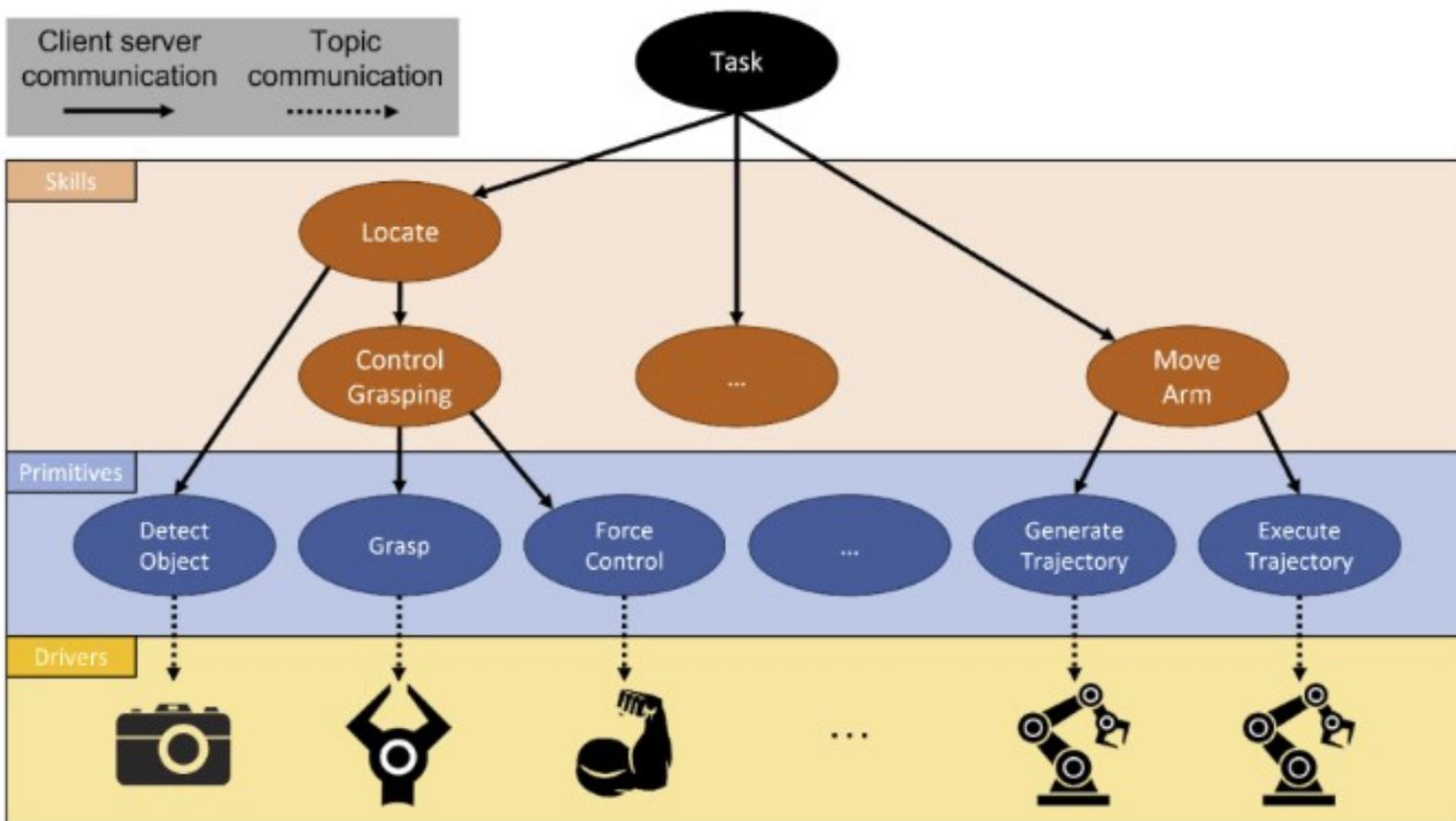
Création d'une tâche d'assemblage à l'aide d'arbre de comportement



 **ROS**

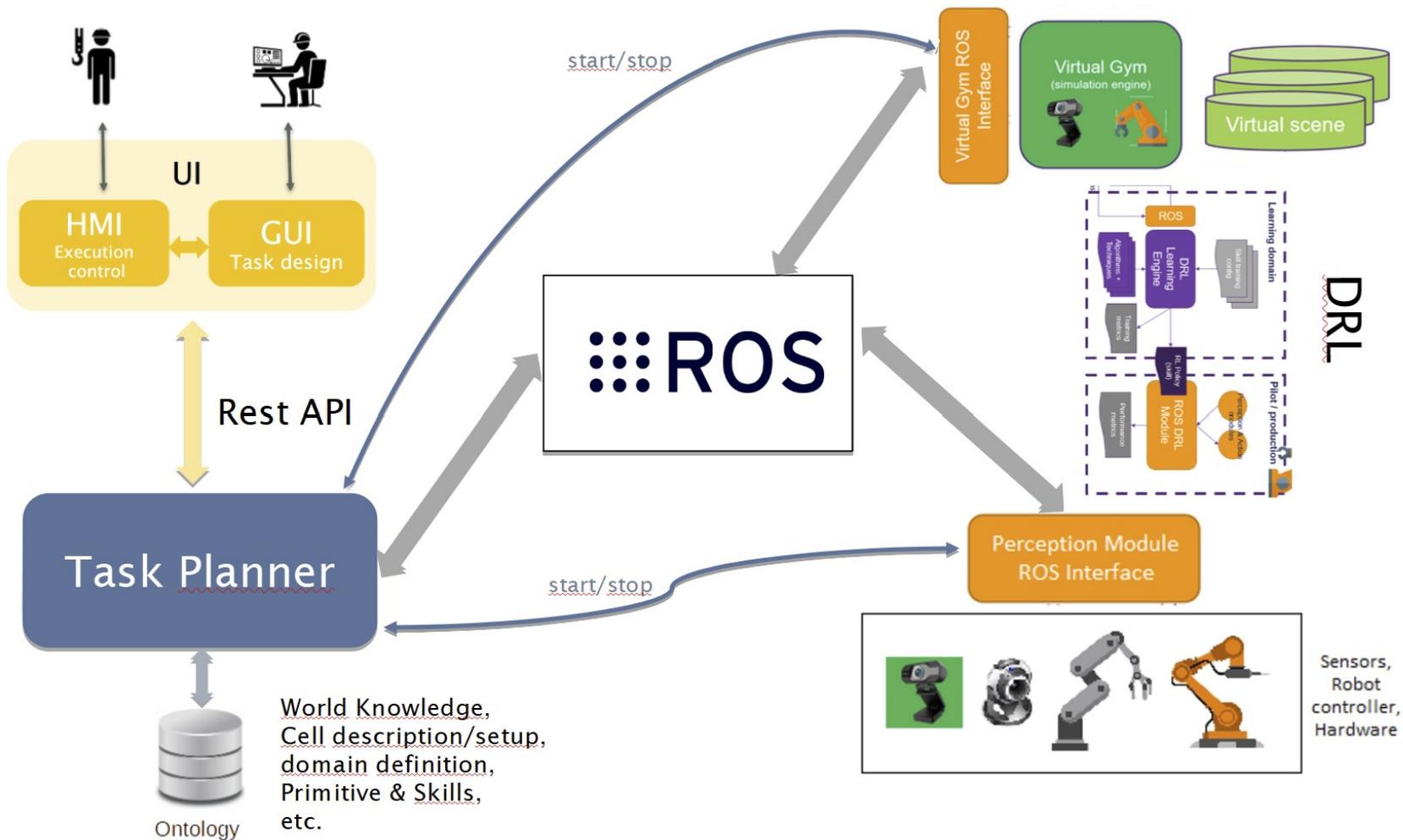
 **docker**

module de perception multi-modale



ROSCON FR Nantes – Workshop 4

Création d'une tâche d'assemblage à l'aide d'arbre de comportement



Tool Bar

Processes

- Human Tasks
 - Fill up Bin
 - Prepare WS
- HRC Tasks
 - Bin Picking
 - Assembly
- Robot Tasks
 - Pick & Place
 - Pick & Place
- Skills
 - Move to
 - Locate
- Primitives
 - Grasp
 - Release
- Connectors
 - → ◇ ○

Working Area

HRC Task – Bin Picking

```

    graph LR
      A[Fill up Bin] --> B[Pick & Place]
  
```

Robot Task – Pick & Place

```

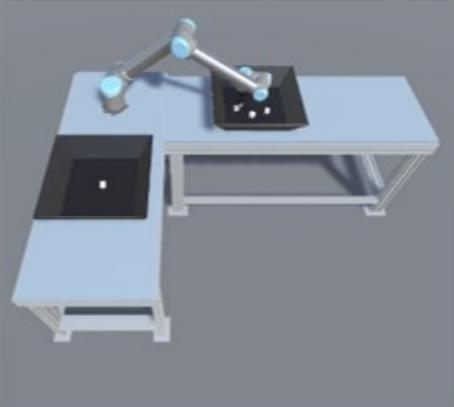
    graph LR
      A[Move to] --> B[Grasp] --> C[Move to] --> D[Release]
  
```

Skill – Move to

```

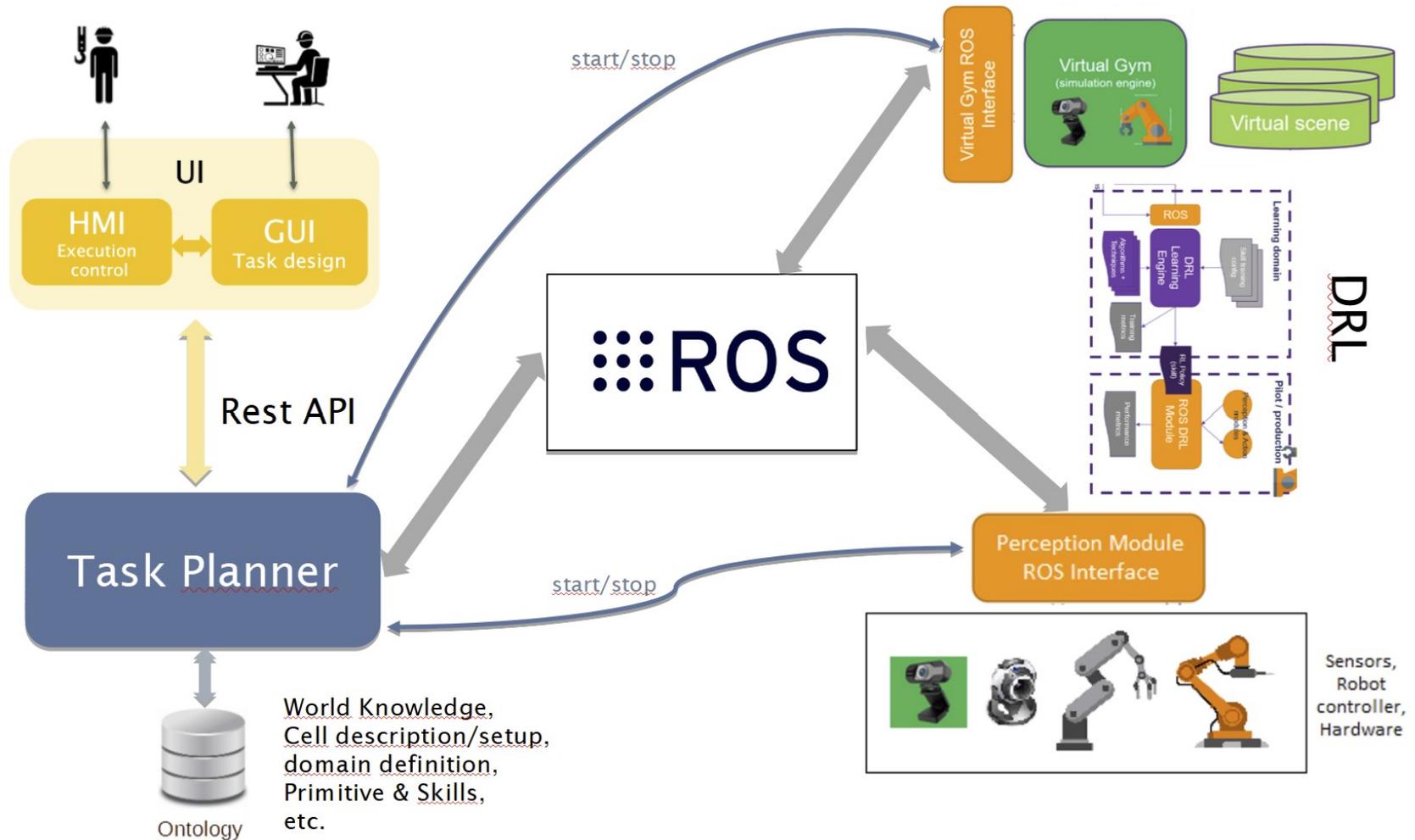
    graph LR
      A[Generate Trajectory] --> B[Execute Trajectory]
  
```

Simulation



ROSCON FR Nantes – Workshop 4

Création d'une tâche d'assemblage à l'aide d'arbre de comportement



Task Planner

- Générer une tâche à partir de la représentation UI et des compétences/primitives existantes.
- Effectuer l'exécution/le contrôle de la tâche.
- Replanification automatique en ligne.

Task Planner

- Générer une tâche à partir de la représentation UI et des compétences/primitives existantes.
- Effectuer l'exécution/le contrôle de la tâche.
- Replanification automatique en ligne.



PDDL

Task Planner

- Générer une tâche à partir de la représentation UI et des compétences/primitives existantes.

 **Arbres de Comportement**

- Effectuer l'exécution/le contrôle de la tâche.

- Replanification automatique en ligne.

 **PDDL**  **Arbres de Comportement**

2. Les Arbres de Comportement

- Généralisation de la Machine à états finis
- Arbre de noeuds hiérarchiques contrôlant la prise de décision
- Souple et puissant
- Utilisé dans les jeux vidéo pour émuler des comportements intelligents

Les Arbres de Comportement

Référence:

Behavior Trees in Robotics and AI
Michele Colledanchise & Petter Ögren

<https://arxiv.org/abs/1709.00084>

Les Arbres de Comportement

- Un signal "tick" est envoyé à la racine de l'arbre à une fréquence donnée
- Le tick est propagé à travers l'arbre jusqu'à ce qu'il atteigne un nœud feuille
- Un nœud qui reçoit un "tick" renvoie l'un des résultats suivants :
RUNNING
SUCCESS
FAILURE
- Si un nœud a plusieurs enfants, il est responsable de propager le signal à ses enfants.

Les Arbres de Comportement

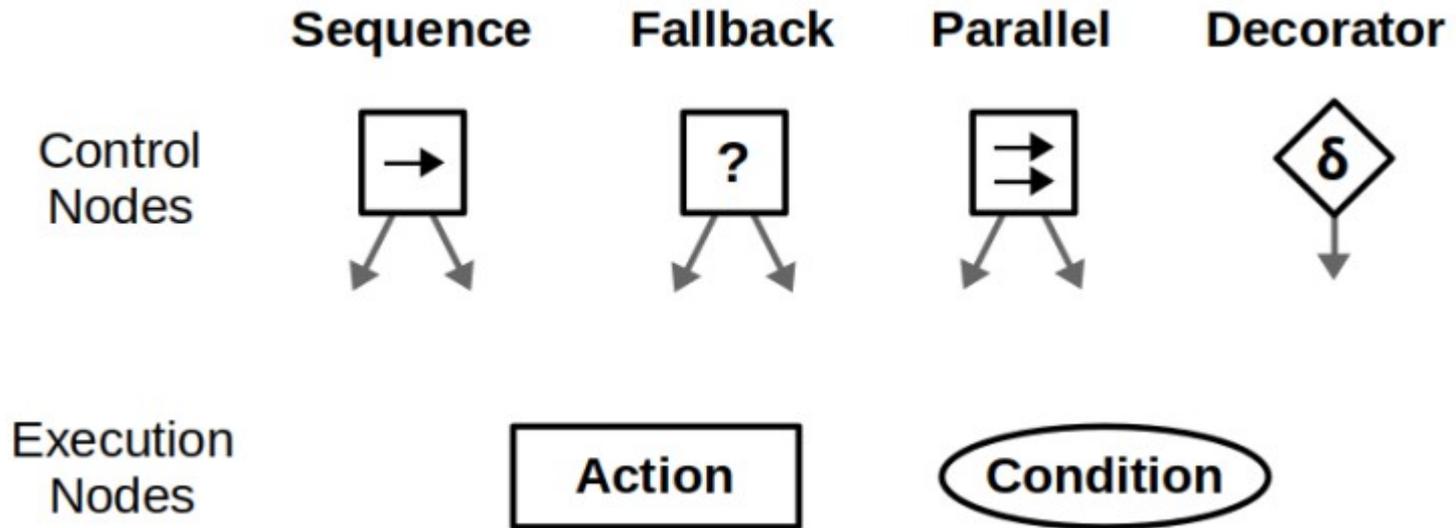
- Les noeuds internes sont des noeuds de "controle"
 - Sequence
 - Fallback
 - Parallel
 - Decorator
- Les noeuds feuilles sont des noeuds d'execution
 - Action
 - Condition

Les Arbres de Comportement

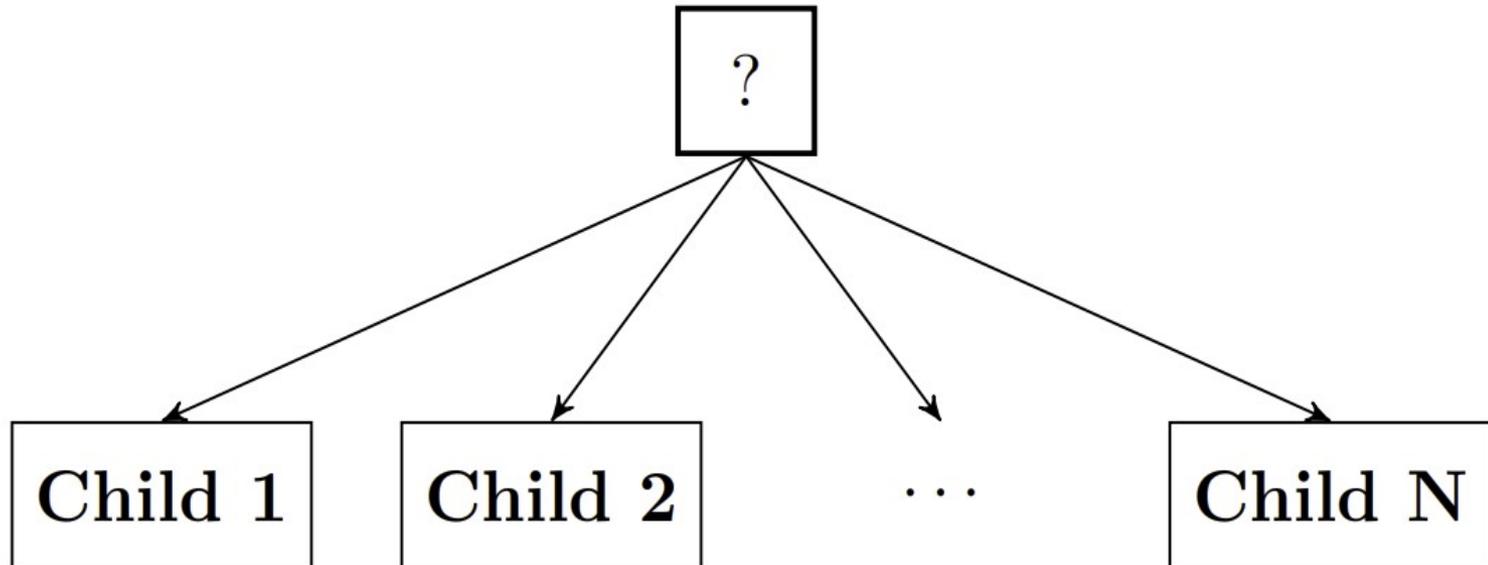
- Les noeuds internes sont des noeuds de "controle"
 - Sequence
 - Fallback
 - Parallel
 - Decorator
- Les noeuds feuilles sont des noeuds d'execution
 - Action
 - Condition

Type of TreeNode	Children Count	Notes
ControlNode	1...N	Usually, ticks a child based on the result of its siblings or/and its own state.
DecoratorNode	1	Among other things, it may alter the result of the children or tick it multiple times.
ConditionNode	0	Should not alter the system. Shall not return RUNNING.
ActionNode	0	This is the Node that "does something"

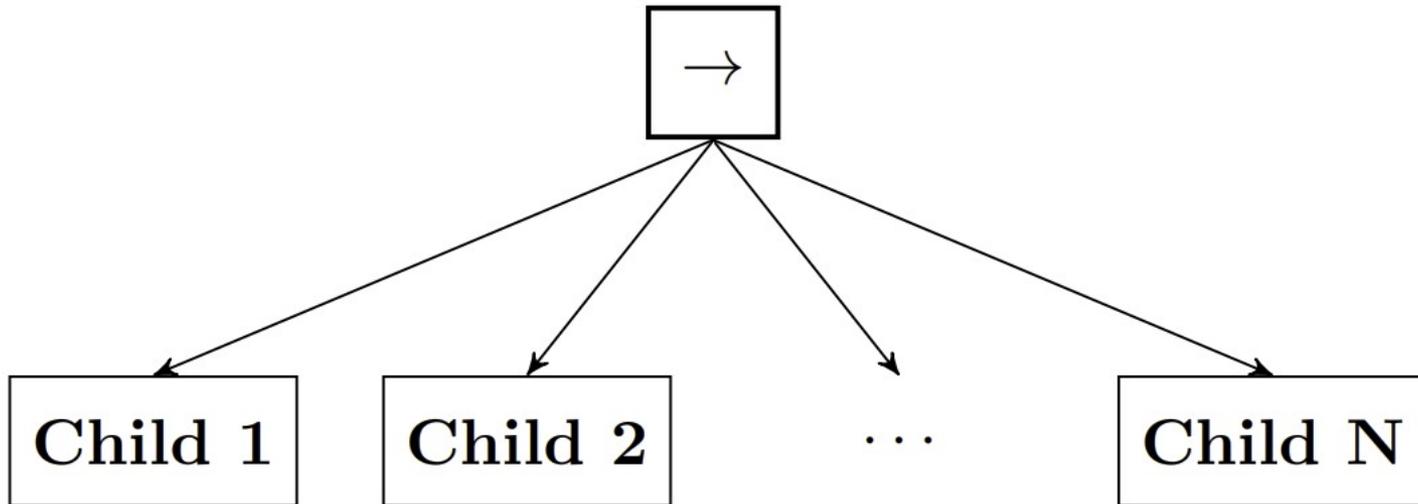
Source: https://www.behaviortree.dev/docs/learn-the-basics/BT_basics



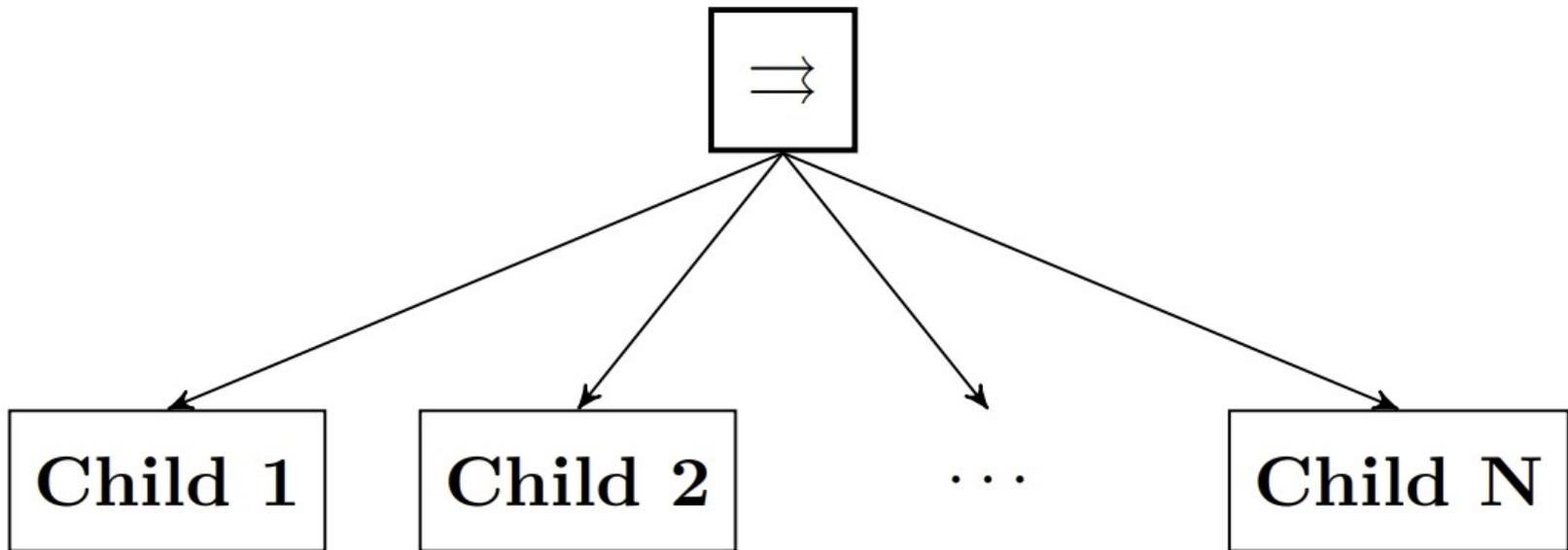
<https://roboticseabass.com/2021/05/08/introduction-to-behavior-trees/>



Source: Behavior Trees in Robotics and AI – Colledanchise & Ögren

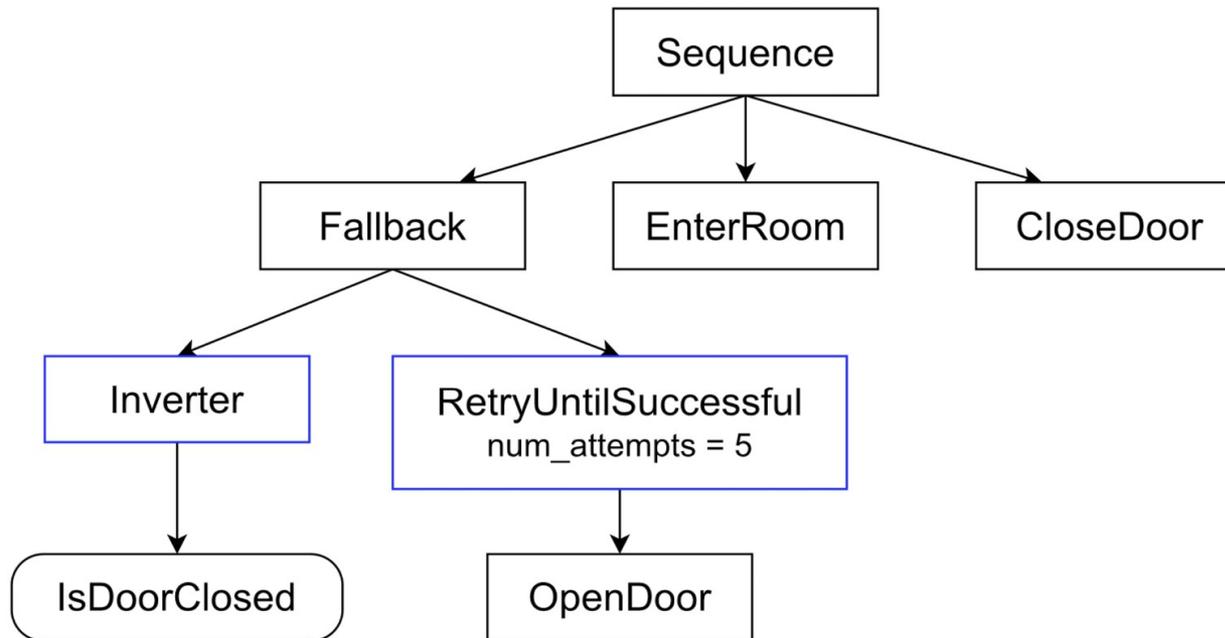


Source: Behavior Trees in Robotics and AI – Colledanchise & Ögren



Source: Behavior Trees in Robotics and AI – Colledanchise & Ögren

Les Arbres de Comportement



Source: https://www.behaviortree.dev/docs/learn-the-basics/BT_basics

Les Arbres de Comportement en bref:

- Simplicité et Réactivité: quelques composants seulement
- Evolutif: pas d'explosion combinatoire lorsque le nombre de noeuds augmentent.
- Dynamique: l'arbre peut être changé "on the fly".
- Ticking: no complex multi-threading.

Les Arbres de Comportement

- Les arbres de comportement modélise le flux de décision
- Flux de données ?

Les Arbres de Comportement

- Les arbres de comportement modélise le flux de décision
- Flux de données ? "Blackboard"

Les Arbres de Comportement

- Les arbres de comportement modélise le flux de décision
- Flux de données ? "Blackboard"
 - une structure de données partagée
 - accessible à tout moment pour écrire ou lire données.

Implémentations

- BehaviorTreeCpp
(<https://www.behaviortree.dev/>)
 - C++
 - Nav2
 - Xml
- py_trees
(<https://py-trees.readthedocs.io/en/devel/>)
 - python

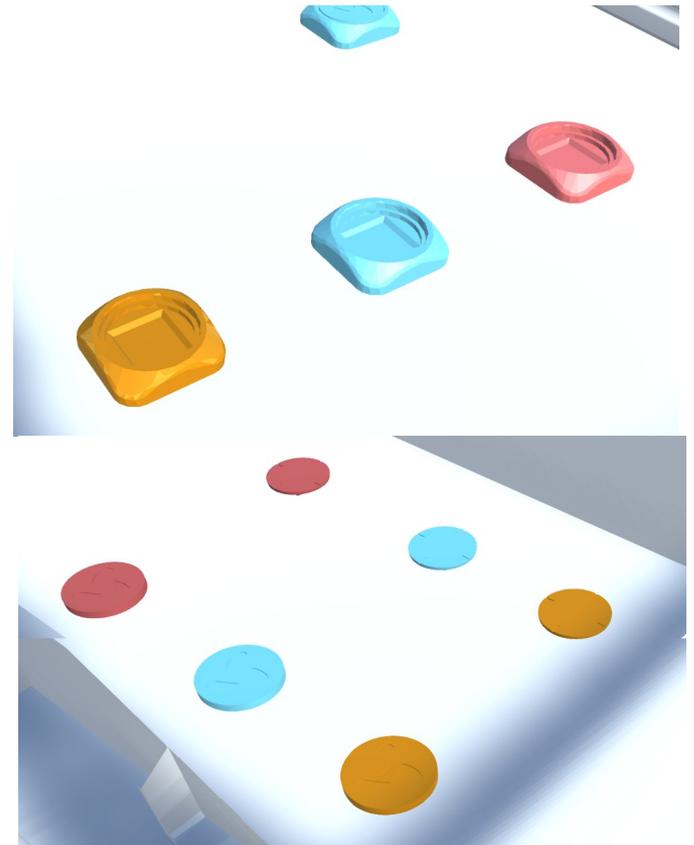
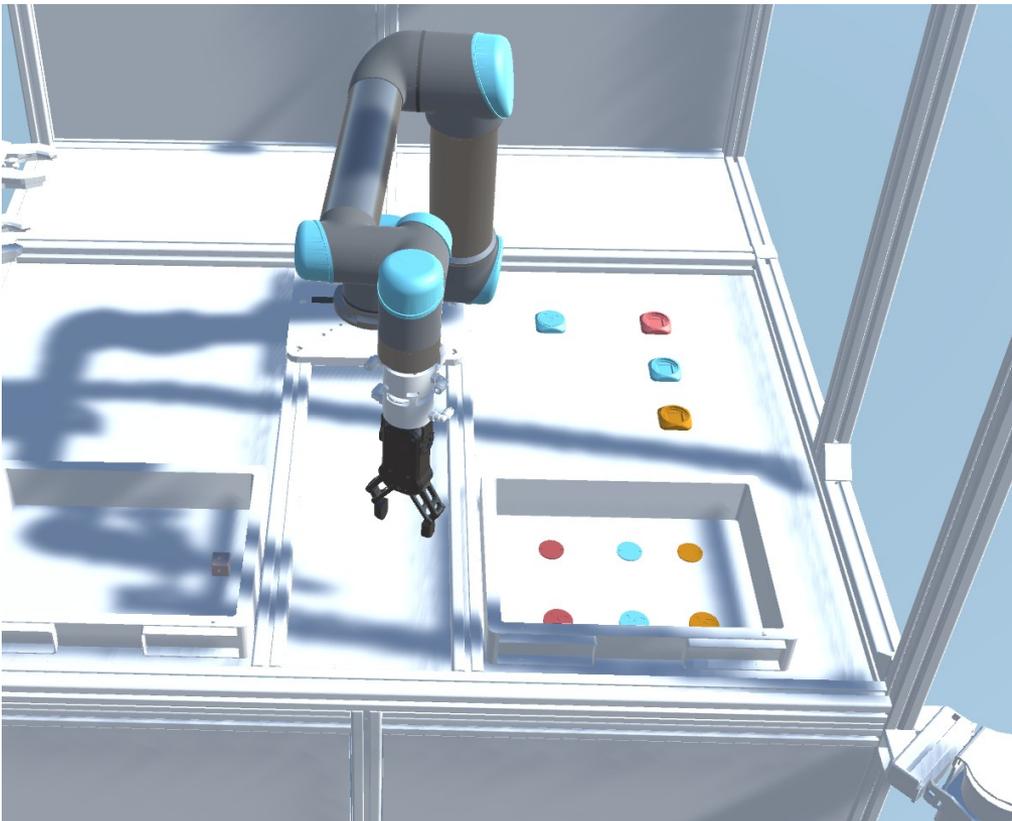
BehaviorTreeCpp

- + performance (C++)
- + representation en XML (modulable, extensible, subtrees)
- + generation pendant l'execution possible (XML)
- + Nav2 implémentation (lifecycle management, plugins)
- + Visualisation Groot
- + Robuste
- C++, BT Wrapper.
- Data flow dans le Blackboard:
 - Système de ports d'entrée et sortie
 - typé statiquement
- Limitations pour la génération pendant l'exécution
- Prise en main complexe/lourde
- + notation et concept sont calqués sur référence BT in AI.

py_trees

- + Python, développement rapide
- + Prise en main rapide
- + Visualisation (Arbre et Blackboard)
- + Debugging facilité
- Pas de représentation intermédiaire
- Performance
- Medium scale
- Notations et concepts "différents"

3. Tache d'assemblage de modèles 3D de montres



Objectif: Assembler des „montres“ de la même couleur

– une version très «épurée» de la plateforme Acroba

- * Environnement de simulation (virtual gym).

- * un nombre très restreint de skills:

 - > MoveTo

 - > OpenGripper

 - > CloseGripper

 - > Locate (sans vision...)

– py_trees

Setup

```
$ git clone git@github.com:rosconfr24-ws4/acroba.git  
$ cd acroba
```

Ou pour la mise à jour du setup existant:

```
$ cd acroba  
$ git pull
```

Dans les deux cas:

```
$ make pull  
$ make setup
```

Pour commencer...

```
$ cd acroba  
$ make run-dev
```

Si WSL2:

```
$ make run-dev VG=WIN
```

Ou

```
$ make run-dev VG=WSL
```

A vous de jouer :)

