UNIVERSITY
OF APPLIED SCIENCES
UPPER AUSTRIA

FELiCE

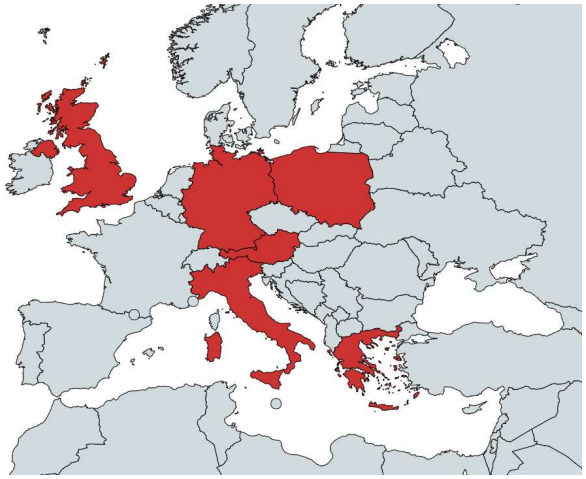
Multi-criteria Optimization of Workflow-based Assembly Tasks in Manufacturing

Florian Holzinger, Andreas Beham

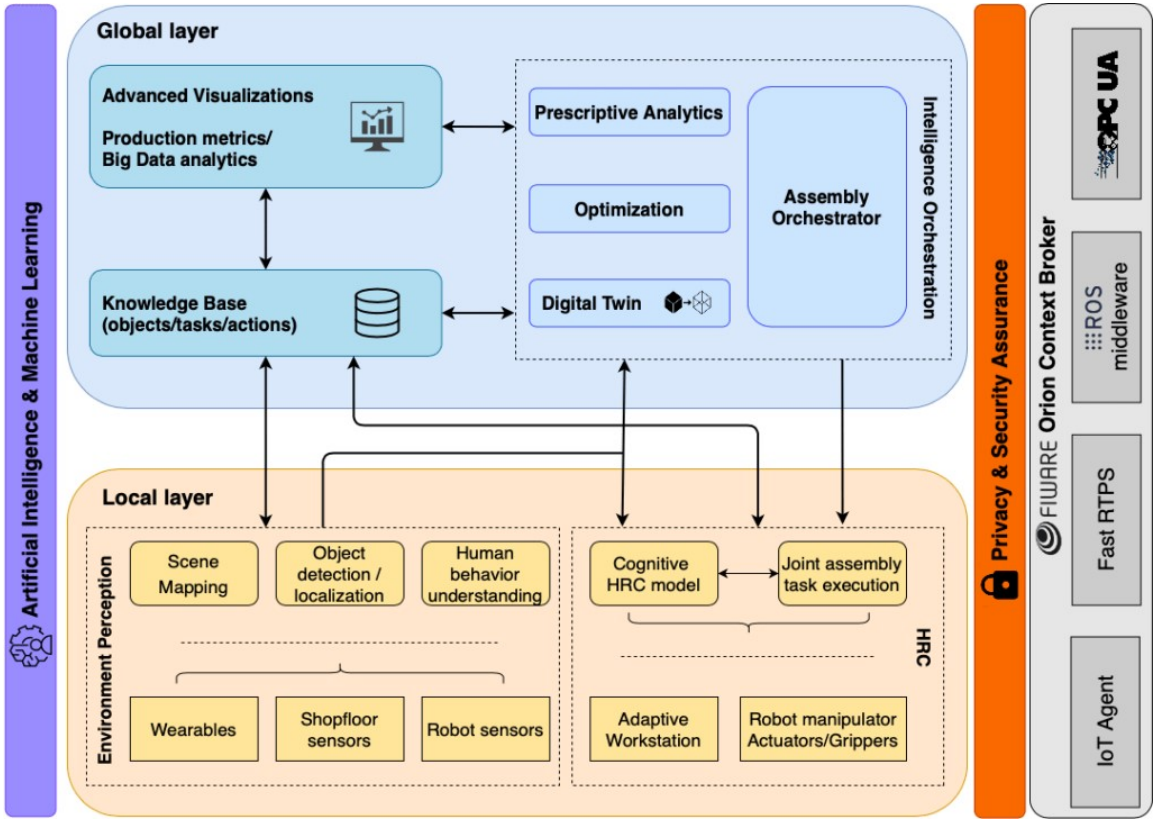
University of Applied Sciences Upper Austria, Hagenberg
Heuristic and Evolutionary Algorithms Laboratory

FLEXIBLE ASSEMBLY MANUFACTURING WITH HUMAN-ROBOT COLLABORATION AND DIGITAL TWIN MODELS

42 Months | 13 Organizations | 6 Countries



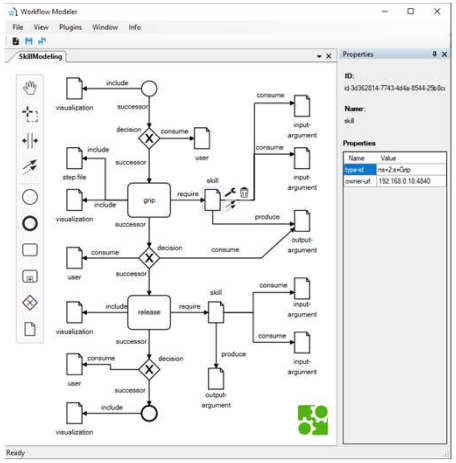
Project overview



Formalization

Implicit knowledge

Explicit knowledge

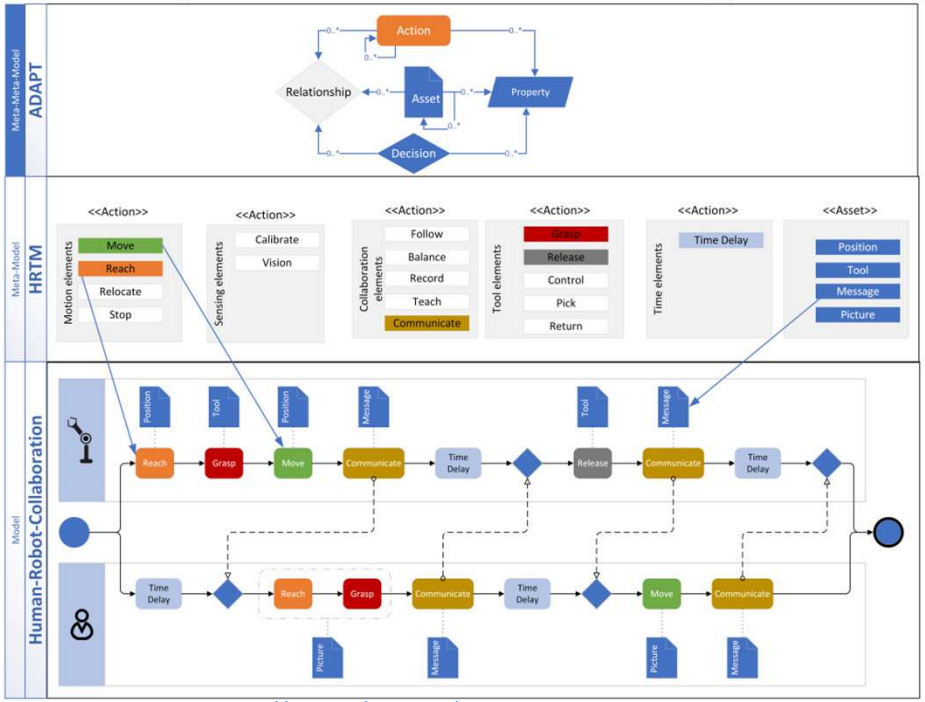


Real world scenarios

Formalisation and Methodology

Formalization

ADAPT - A decision-model-based Approach for Modeling Collaborative Assembly and Manufacturing Tasks

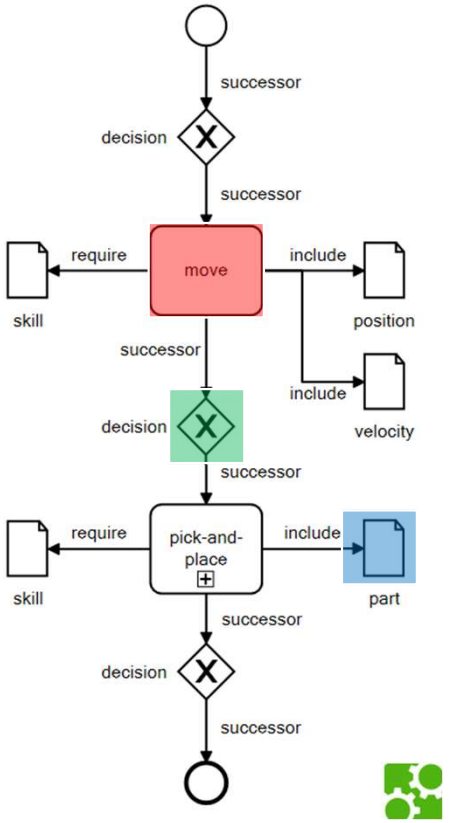


https://doi.org/10.3217/978-3-85125-663-5-14

Formalization

A Workflow consists of:

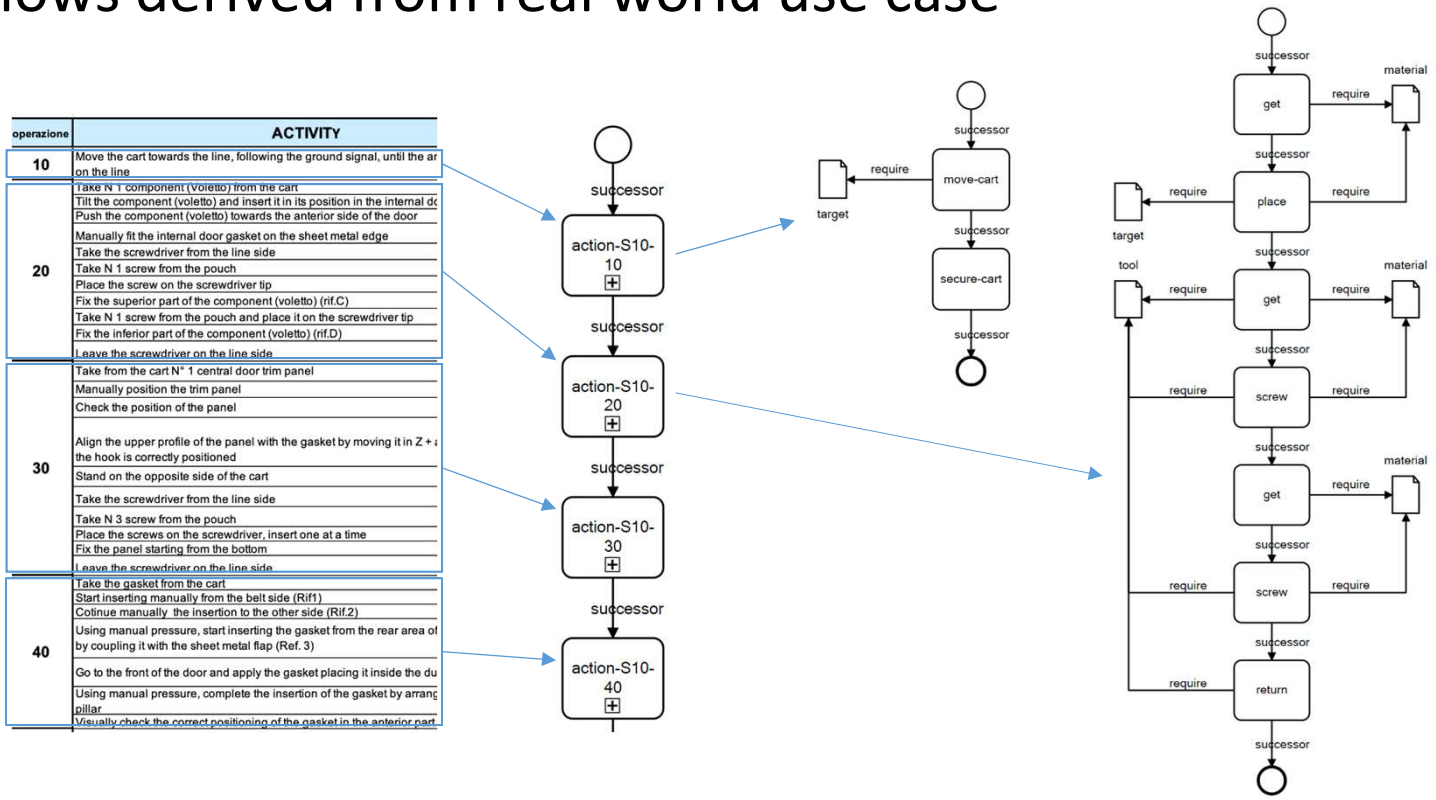
- Actions
  - What has to be done
- Decisions
  - According to environment
- Assets
  - Resources allocated to Actions
- Relationships
  - How everything is connected



Formalization

Workflows derived from real world use case

operazione	ACTIVITY
10	Move the cart towards the line, following the ground signal, until the ar on the line
20	Take N 1 component (Voletto) from the cart
	Tilt the component (voletto) and insert it in its position in the internal dc
	Push the component (voletto) towards the anterior side of the door
	Manually fit the internal door gasket on the sheet metal edge
	Take the screwdriver from the line side
	Take N 1 screw from the pouch
	Place the screw on the screwdriver tip
	Fix the superior part of the component (voletto) (rif.C)
	Take N 1 screw from the pouch and place it on the screwdriver tip
	Fix the inferior part of the component (voletto) (rif.D)
	Leave the screwdriver on the line side
30	Take from the cart N° 1 central door trim panel
	Manually position the trim panel
	Check the position of the panel
	Align the upper profile of the panel with the gasket by moving it in Z + a the hook is correctly positioned
	Stand on the opposite side of the cart
	Take the screwdriver from the line side
	Take N 3 screw from the pouch
	Place the screws on the screwdriver, insert one at a time
	Fix the panel starting from the bottom
	Leave the screwdriver on the line side
40	Take the gasket from the cart
	Start inserting manually from the belt side (Rif1)
	Cotinue manually the insertion to the other side (Rif.2)
	Using manual pressure, start inserting the gasket from the rear area of by coupling it with the sheet metal flap (Ref. 3)
	Go to the front of the door and apply the gasket placing it inside the du
	Using manual pressure, complete the insertion of the gasket by arrang pillar
	Visually check the correct positioning of the gasket in the anterior part

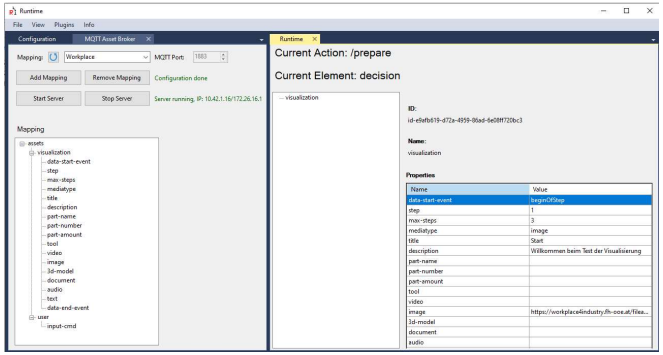


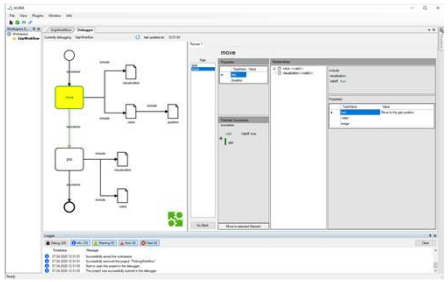
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Formalization

Supported by a variety of software products

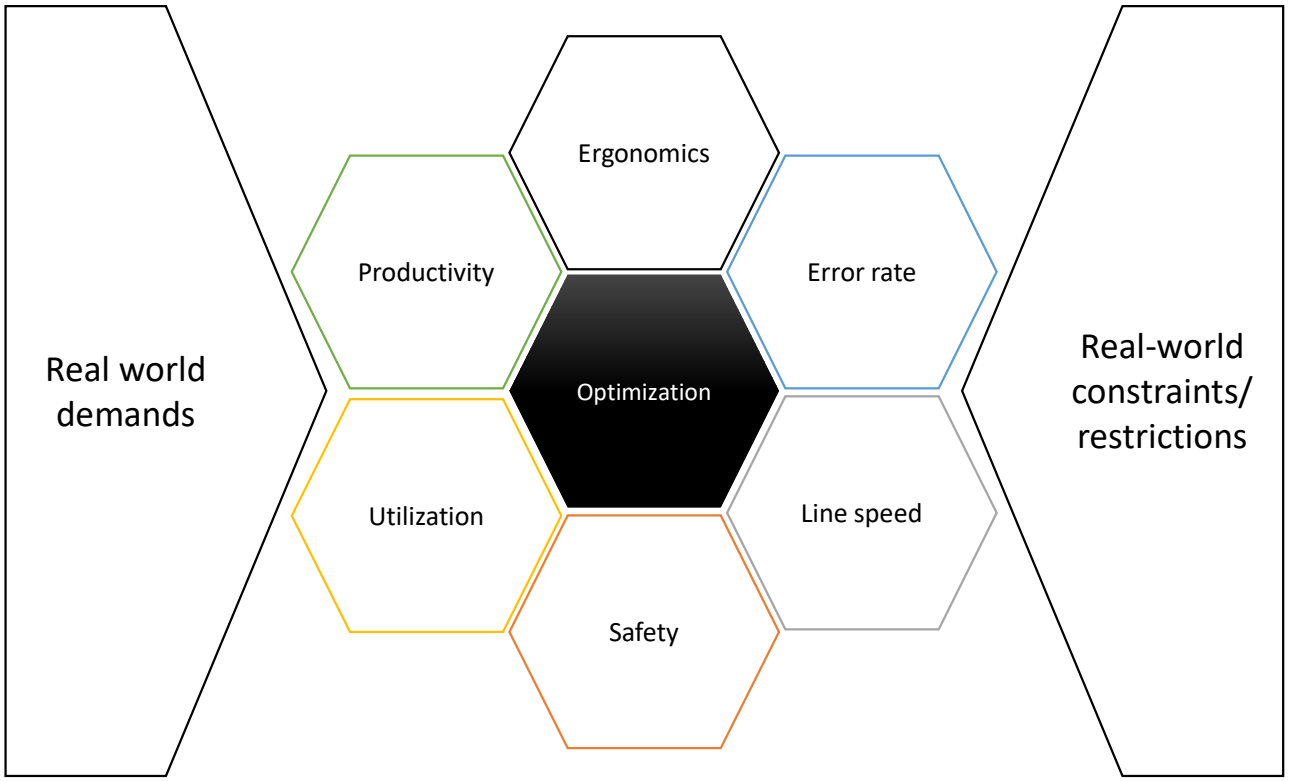
- Workflow Modeler (WORM)
- Runtime
- Visualization



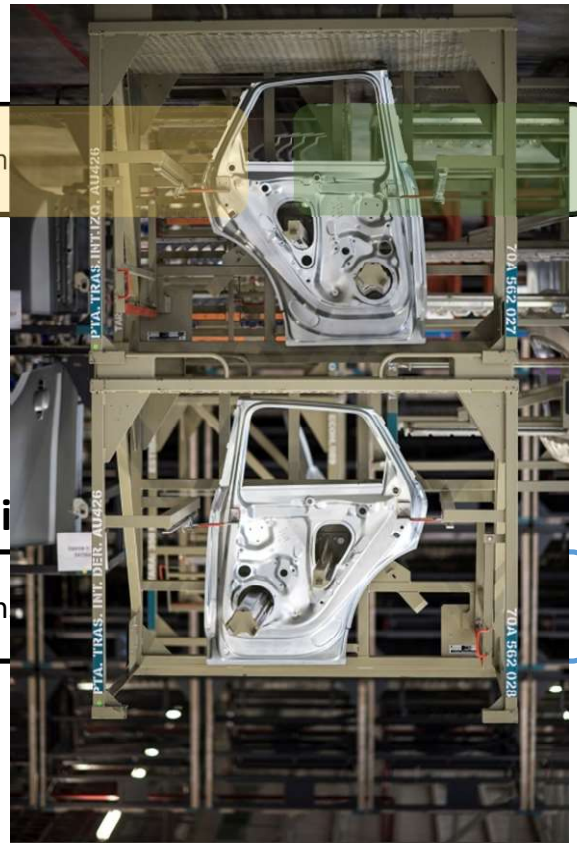


https://sar.fh-ooe.at/index.php/de/downloads

Optimization of Workflows



Optimization of Workflows (Ergonomics)



Workflow (Human only)

Take the speaker from the cart → In → Fix 4 screws

3 Penalty + + 2 Penalty = 8 Penalty

Workflow (Cobot utilizati

Take the speaker from the cart → In → Fix 4 screws

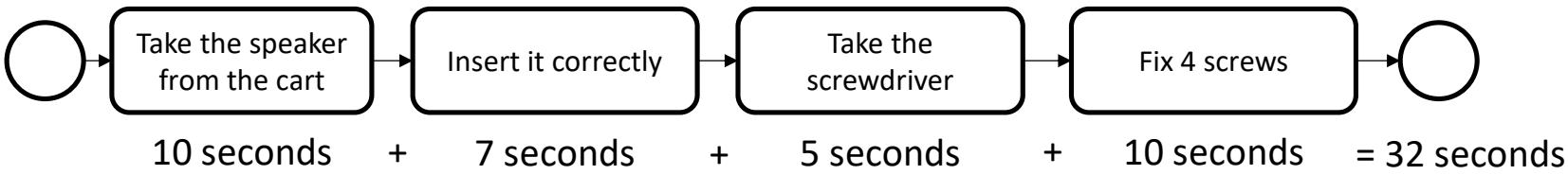
~~3 Penalty~~ + 2 Penalty = 4 Penalty

Done by human

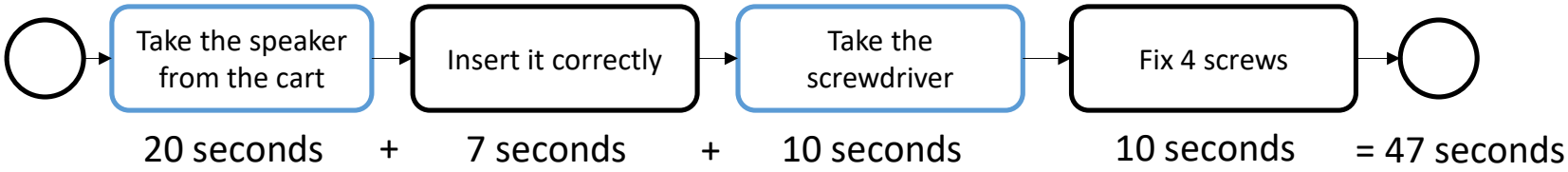
Done by robot

Optimization of Workflows (Makespan)

Workflow (Human only)



Workflow (Cobot utilization)

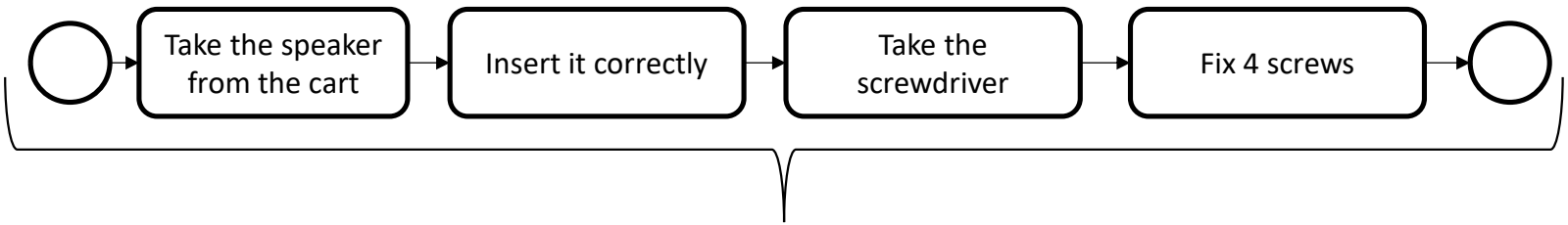


Done by human

Done by robot

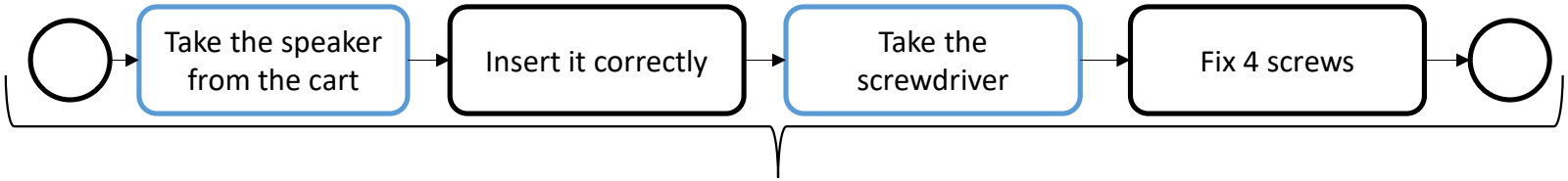
Optimization of Workflows (Multi-objective)

Workflow (Human only)



Ergonomics: 8 Penalty Makespan: 32 seconds
Criteria C: x_1 units Criteria D: y_1 units
…

Workflow (Cobot utilization)



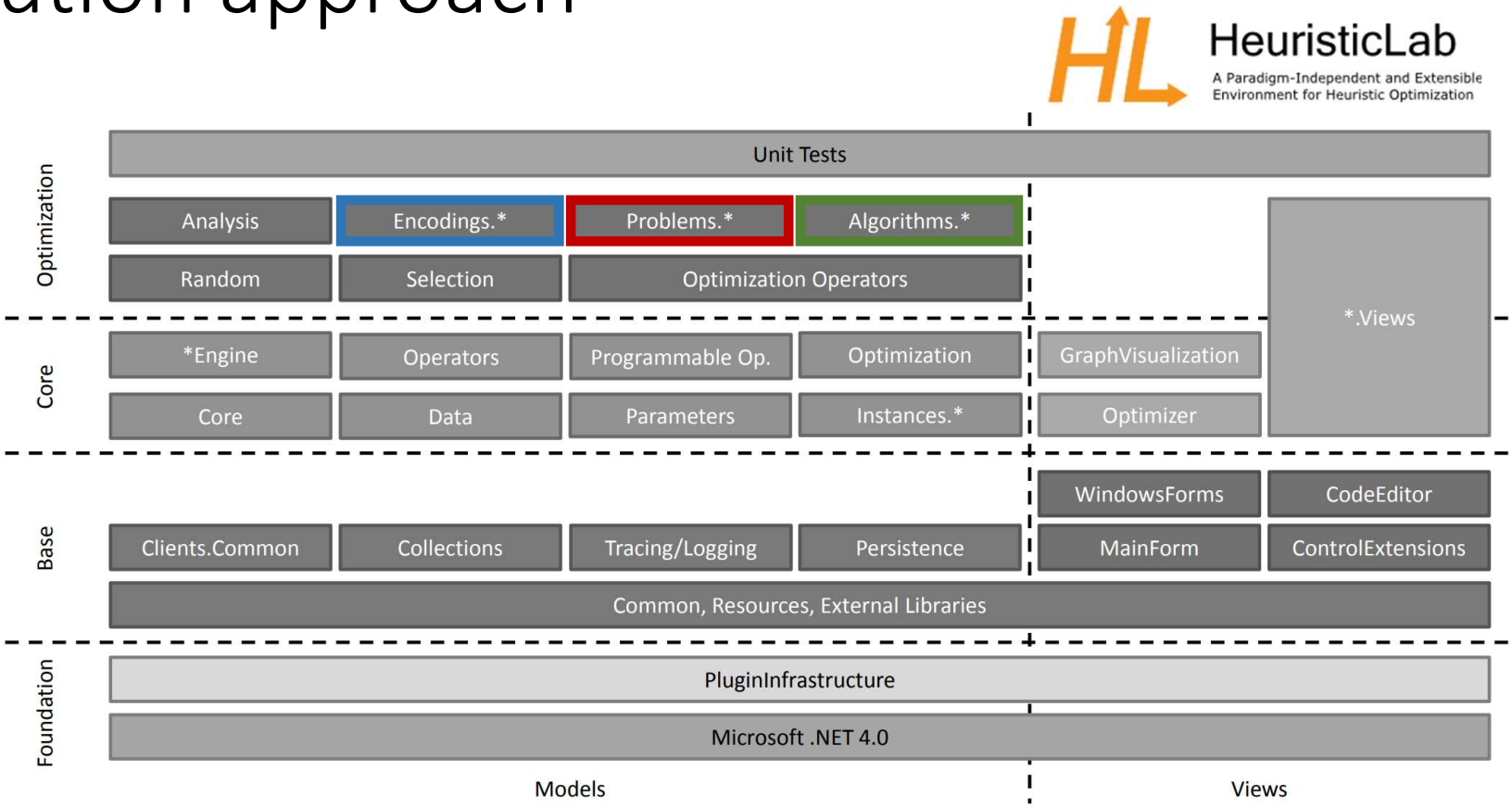
Ergonomy: 4 Penalty Makespan: 47 seconds
Criteria C: x_2 units Criteria D: y_2 units
…

Done by human
Done by robot

Problem

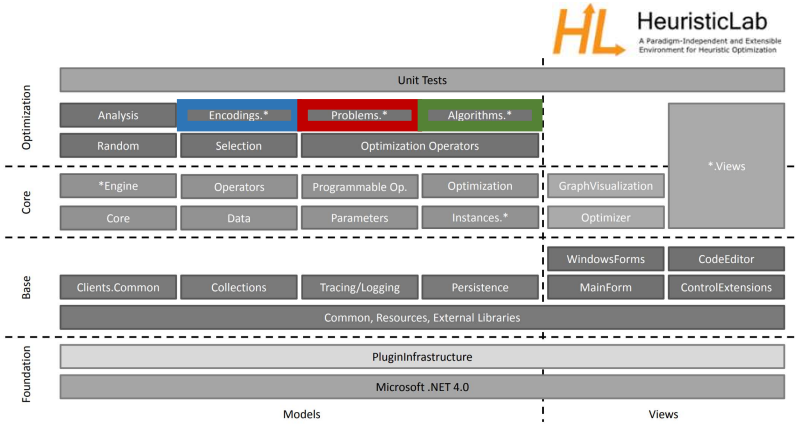
- Multiple, sometimes rivalling optimization criteria
  - Requiring additional meta-information about workflow
- Depended on scenario/use-case
  - And of course the available elements (cobot/adaptive workstation, …)
- Requires both domain expert and optimization expert knowledge
- Requirements will change over time
  - New/changing optimization criteria
- Currently no optimization framework/approach for ADAPT exists

Solution approach



DOI: 10.1007/978-3-319-01436-4_10

Solution approach



1. Define new optimization problem

2. Utilize available multi-objective algorithms
   - NSGA-II / NSGA-III

3. Exploit MultiEncoding
   - Allows employing different encodings (BinaryVector, Permutation, …)

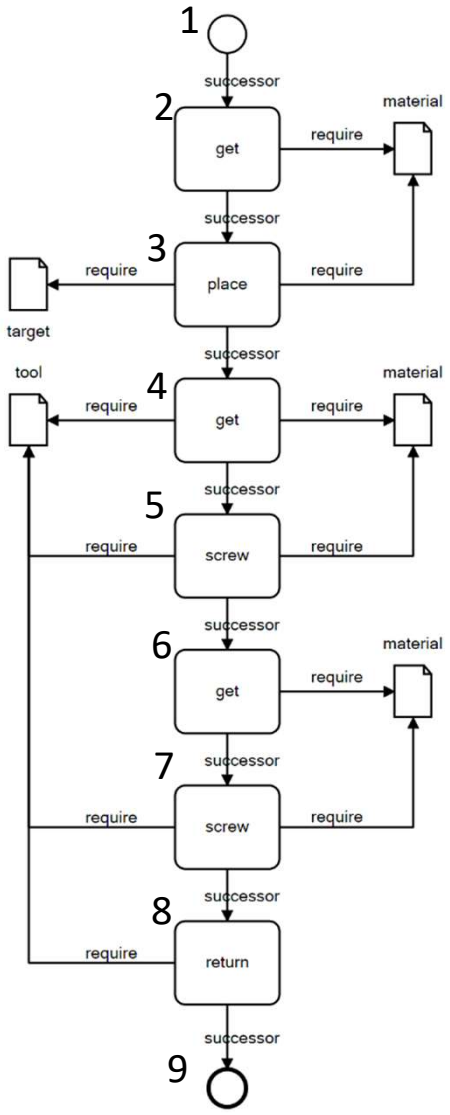
DOI: 10.1007/978-3-319-01436-4_10

Solution workflow

1. Load workflow (including meta model)
2. Enumerate action nodes
3. Append meta-information
   1. Meta workflow adaptations
   2. Append meta-information in workflow properties
4. Initialize multi-objective optimization algorithm with MultiEncoding
5. Let the metaheuristic guide the search process
   1. Utilizing a predefined set of Manipulations
   2. Calculate an arbitrary number of fitness functions

Properties

Name	Value
torque	10
optimalHeight	120
precedences	get_2
ergonomicImpact	2
approxTime	5



UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Solution – A quick glance at the code

IMetaInformationAppender
Interface

- Properties
  - *Description { get; } : string*
  - *Name { get; } : string*
- Methods
  - *AppendMetaInformation(Workflow workflow) : void*

IPrimitiveFitnessCalculator
Interface

- Properties
  - *IsMaximization { get; } : bool*
- Methods
  - *CalculateFitness(Workflow workflow) : double*
  - *CheckPrecondition(Workflow workflow) : bool*

IWorkflowManipulator<T>
Generic Interface

- Properties
  - *Description { get; } : string*
  - *Name { get; } : string*
- Methods
  - *ManipulateGraph(Workflow workflow, Individual individual, Dictionary<int, string> nodeMapping) : void*

IComplexFitnessCalculator
Interface

- Properties
  - *IsMaximization { get; } : List<bool>*
- Methods
  - *CalculateFitness(Workflow workflow) : List<double>*
  - *CheckPrecondition(Workflow workflow) : bool*

Examples – I (Cobot utilization)

```
public class CRFCobotSkillMetaInformationAppender : IMetaInformationAppender
{
    private HashSet<string> cobotSkills = new HashSet<string>() { "screw", "grip" };

    public static string PROPERTYNAME_SKILLAVAILABLE = "CobotSkillAvailable";
    public static string PROPERTYNAME_COBOTUTILIZED = "CobotUtilized";

    public string Name => nameof(CRFCobotSkillMetaInformationAppender);
    public string Description => "Adds information about cobot skills and utilization";

    public void AppendMetaInformation(Workflow workflow)
    {
        workflow.container.GetMetaModel().GetActionTypes().ForEach(v =>
        {
            v.PropertyTypes.Add(new PropertyType(PROPERTYNAME_SKILLAVAILABLE));
            v.PropertyTypes.Add(new PropertyType(PROPERTYNAME_COBOTUTILIZED));
        });

        foreach (var action in workflow.container.GetActions())
        {
            bool cobotSkillAvailable = cobotSkills.Contains(action.TypeName.ToLower());
            action.Properties.Add(new Property(PROPERTYNAME_SKILLAVAILABLE, cobotSkillAvailable.ToString()));
        }
    }
}
```

Examples – I (Cobot utilization)

```
public class CRFCobotUtilizationWorkflowManipulator : IWorkflowManipulator<BinaryVectorEncoding>
{
    public string Name => nameof(CRFCobotUtilizationWorkflowManipulator);

    public string Description => "Adjusts cobot utilization";

    public void ManipulateGraph(Workflow workflow, Individual individual, Dictionary<int, string> nodeMapping)
    {
        var vectorEncoding = individual.BinaryVector();
        for (int i = 0; i < vectorEncoding.Length; i++)
        {
            var action = workflow.container.GetAction(nodeMapping[i]);
            bool skillAvailable = action.GetProperty(CRFCobotSkillMetaInformationAppender.PROPERTYNAME_SKILLAVAILABLE).Value == "true";
            bool skillUtilized = vectorEncoding[i] && skillAvailable;

            action.GetProperty(CRFCobotSkillMetaInformationAppender.PROPERTYNAME_COBOTUTILIZED).Value = skillUtilized.ToString();
        }
    }
}
```

Examples – I (Cobot utilization)

```
public class CRFCobotUtilizationErgonomicsFitnessCalculator : IPrimitiveFitnessCalculator
{
    public bool IsMaximization => false;

    public double CalculateFitness(Workflow workflow)
    {
        double ret = 0;
        foreach(var action in workflow.container.GetActions())
        {
            ret += CalculateErgonomics(action);
        }

        return ret;
    }

    public bool CheckPrecondition(Workflow workflow)
    {
        return workflow.container.GetMetaModel().GetActionTypes().All(v =>
        {
            return v.GetPropertyTypeNames().Contains(CRFCobotSkillMetaInformationAppender.PROPERTYNAME_COBOTUTILIZED);
        });
    }

    public double CalculateErgonomics(Action action)...
}
```

Examples – I (Cobot utilization)

```
public class CRFCobotUtilizationComplexFitnessCalculator : IComplexFitnessCalculator
{
    public List<bool> IsMaximization => new List<bool>() { false, false };

    CRFWorkflowSimulation sim = new CRFWorkflowSimulation(defaultInitialization: true);

    public List<double> CalculateFitness(Workflow workflow)
    {
        return sim.Simulate(workflow);
    }

    public bool CheckPrecondition(Workflow workflow)
    {
        return workflow.container.GetMetaModel().GetActionTypes().All(v =>
            v.GetPropertyTypeNames().Contains(CRFCobotSkillMetaInformationAppender.PROPERTYNAME_COBOTUTILIZED)
        );
    }
}
```

Summary

- Created use-case agnostic framework for multi-criteria optimization of ADAPT workflows by utilizing HeuristicLab
  - Designed as new HeuristicLab plugin
  - Enables utilization of existing multi-objective optimization algorithms
  - Allows different encodings

UNIVERSITY
OF APPLIED SCIENCES
UPPER AUSTRIA

FELiCE

Multi-criteria Optimization of Workflow-based Assembly Tasks in Manufacturing

Florian Holzinger, Andreas Beham

University of Applied Sciences Upper Austria, Hagenberg
Heuristic and Evolutionary Algorithms Laboratory