Aalen University

Intelligent Task Assignment in Industry 4.0 Production Processes Utilizing Fuzzy Sets

Gregor Grambow, Daniel Hieber, and Roy Oberhauser
Aalen University
Germany

Presenter: Roy Oberhauser

- Worked for 14 years in the software industry in the Silicon Valley and in Germany doing research and development.
- Since 2004 he has been a Professor of Computer Science at Aalen University in Germany, teaching in the areas of software engineering.
- His research interest is to leverage technologies and techniques to innovate, automate, support, and improve the production and quality of software for society.

Contents

Current BPMS Staff Assignment Rules

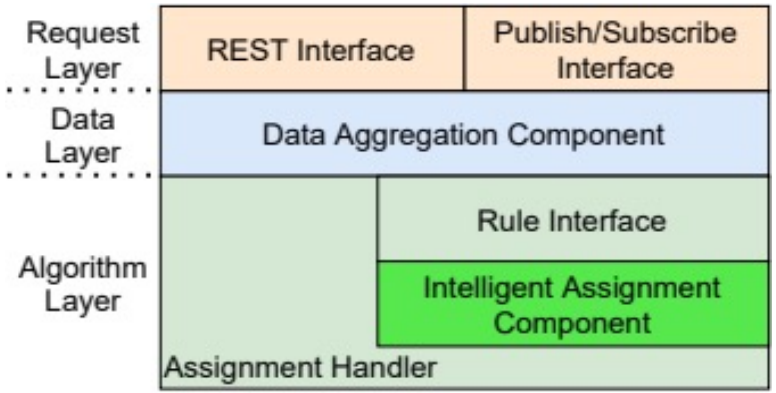
- Staff Assignment Rules (SAR) are used in Business Process Management (BPM) systems (BPMSs) to assign human resources for manual tasks in processes
  - Often utilizing chaining
  - Can determine if agent is capable of task but not the degree of suitability
- Limited custom BPMS-specific solutions hamper access and integration / utilization of more intelligent solutions
- Limited support for leveraging external information during decision making

Solution: Intelligent Assignment Component (IAC)

- Fuzzy Logic provides a granular suitability score
  - Determine if an agent is capable
  - Determine how capable an agent is
- External service allows BPMS-independent implementation & integration
- Inclusion of external information during assignment calculation

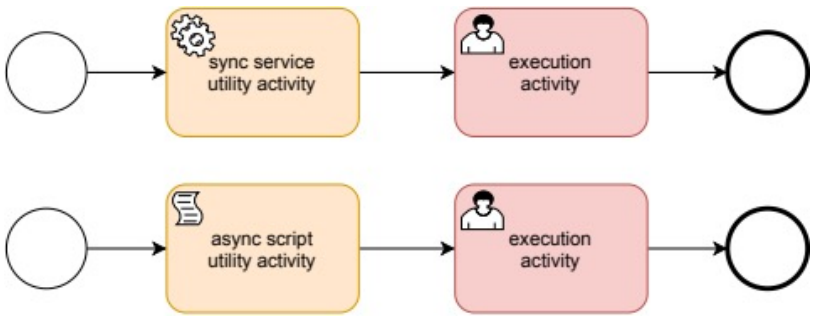
Conceptual Architecture

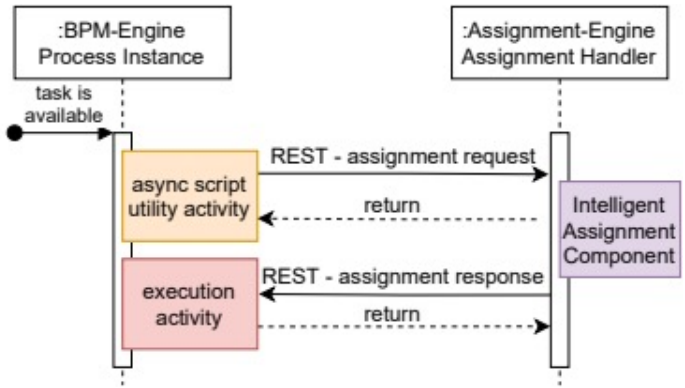
- Request Layer
  - REST for assignment Requests
  - Pub/Sub for Data Aggregation (primarily for sensor data)
- Data Layer
  - Data Aggregation Component aggregates context information missing in initial assignment request
- Algorithm Layer
  - Assignment Handler
    - Coordinates parallel multiprocessing
    - Integration of rule engines for handling preconditions
    - Manages the Intelligent Assignment Component which uses Fuzzy Logic



Intelligent Assignment Component Integration in BPM Process

- Integration of Intelligent Assignment Component via BPMN 2.0 elements
- Easy integration via an additional “Utility“ activity
- No changes to BPMS required
- Intelligent Assignment Component (IAC) can be directly integrated in Service Task or Script Task
- Assignment Engine’s REST Interface can be called synchronously by a service task
- Assignment Engine’s REST Interface can be called asynchronously by a script task
- Optional: Staff Assignment Rules are validated



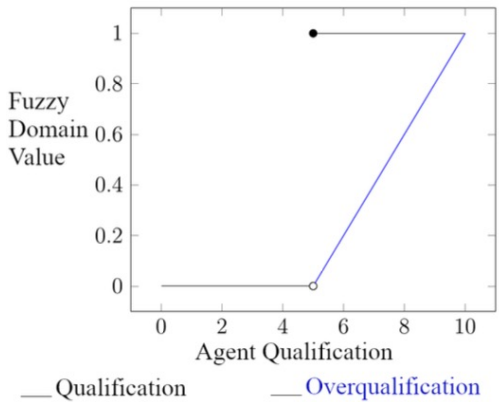


Intelligent Assignment Component

- Assessment Criteria Model (customizable)
  - Distance (travel)
  - Qualification
  - Hourly Rate
  - Workload
  - Danger Level
- Activity Model
- Agent Model
- Assignment Algorithm
  - Exclusion criteria: prevent further calculation
    - Under-qualification
    - Overload (workload)
    - Danger threshold exceeded (worker certification)
  - Fuzzy calculation
    - Assignment score (highest = most suitable)

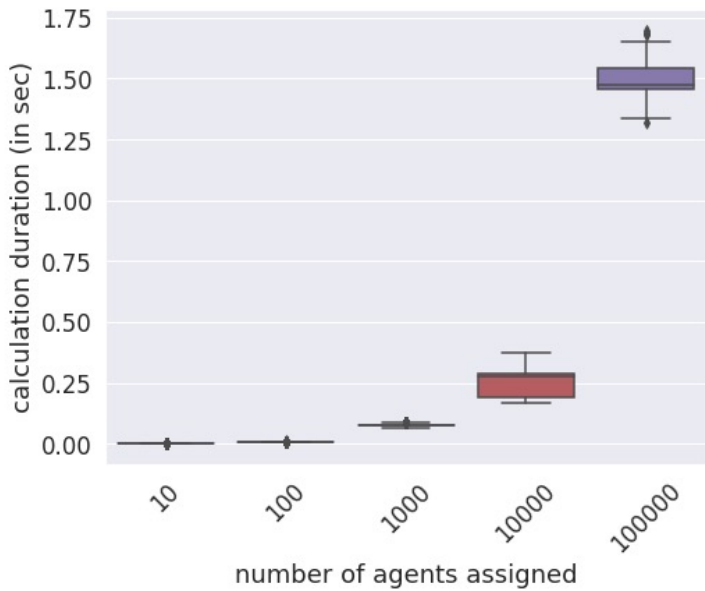
Fuzzy implementation: code snippet for agent qualification

```
def eval_qualification(required: dict, values: dict):
    """eval qualification and over qualification for tasks
    required value and agents value"""
    qualifications = {"types": {}}

    for qualification_type, required_qualification in required.items():
        Qualification = Domain("qualification", 0, 10, res=1)
        qualified = (
            rectangular(required_qualification, 10)
            if required_qualification < 10
            else singleton(required_qualification)
        )
        over_qualified = (
            Set(R(required_qualification, 10))
            if required_qualification < 10
            else Set(constant(0))
        )
        begin_flat = Set(singleton(required))

        if required_qualification != 0:
            begin_flat = MAX(
                ~Set(rectangular(0, required_qualification)),
                Set(singleton(required_qualification)),
            )
        Qualification.qualified = qualified
        Qualification.over_qualified = over_qualified
        Qualification.not_over_qualified = (
            Set(S(required_qualification, 10))
            if required_qualification < 10
            else Set(begin_flat)
        )

        if required_qualification != 0 and required_qualification != 10:
            Qualification.not_over_qualified = product(
                begin_flat, Set(S(required_qualification, 10))
            )
        if required_qualification == 10:
            Qualification.not_over_qualified = begin_flat

        qualifications["types"][qualification_type] = float(
            Qualification.min(values[qualification_type])
        )

    Qualification = Domain("qualification", 0, 1, res=0.001)
    weight = {"electric": 0.25, "computer": 0.25, "social": 0.25, "bio_chemical": 0.25}

    w_func = weighted_sum(weights=weight, target_d=Qualification)
    qualifications["weighted"] = w_func(qualifications["types"])

    return qualifications
```



Performance Evaluation

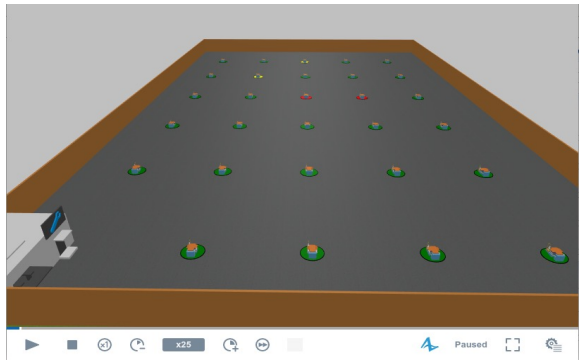
- Evaluation regarding performance and feasibility
- Calculated linear runtime O(n) confirmed
- Multiprocessing provides speedup with large amount of agents (>1000 in single assignment)



Integration Evaluation

- Comparison against default SAR
- Simulated repair and maintenance scenario using AnyLogic
  - Factory with 21504m^2 so that travel distance makes a difference
  - 29 machines requiring maintenance every 16 hours; timespan 36 hrs.
- Easy integration in BPMS with REST interfaces (Camunda, AristaFlow)
- Camunda (CMD) Setup:
  - Agents fetched activities directly from BPMS. Available activity is claimed by any available agent
- IAC Setup:
  - Agents check personal worklist at the Assignment-Engine's REST API. If personal worklist contains activity, they work on it, otherwise idle.

Reduced work time for workers & travel distance
Heavily reduced cost through more efficient assignments



	IA	Camunda
total_activities (amount)	13.98	16.84
work_time (in minutes)	1636.38	1955.20
idle_time (in minutes)	523.62	204.80
cost (in €)	10.00	420.00
avg_overqualification (value)	0.34	0.09
max_avg_under-qualification (value)	0.00	-0.02
traveled_dist (in meters)	7346.92	8911.79
downtime_maintain (in minutes)	484.18	303.30
downtime_repair (in minutes)	204.00	138.23

	IA-int	IA-ext	CMD-int	CMD-ext
total_activities	17.38	0.40	16.85	16.80
work_time	2037.25	32.88	1946.62	1989.50
idle_time	122.75	2127.12	213.38	170.50
cost	0.00	10.00	0.00	420.00
avg_overqual	0.05	1.50	0.06	0.20
max_avg_uqual	0.00	0.00	-0.02	0.00
traveled_dist	9082.78	403.45	8750.17	9558.25

Conclusion

- Performant and accurate task assignments
- Highly adjustable
- Fuzzy Logic as core technology
- Respect labor safety and possible preconditions as part of task assignment
  - No manual checks required
  - Task is only assigned if it is save and ready to perform