FedCSIS 2012
FEDERATED CONFERENCE ON COMPUTER SCIENCE AND INFORMATION SYSTEMS
SEPTEMBER 9 - 12
WROCŁAW, POLAND

AGH

AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY

# Ontology Based Integration and Decision Support in the Insigma Route Planning Subsystem

**Piotr Szwed, Piotr Kadłuczka, Wojciech Chmiel, Andrzej Głowacz,**

**AGH University of Science and Technology**

**Joanna Śliwa**

**Military Communication Institute**

## Agenda

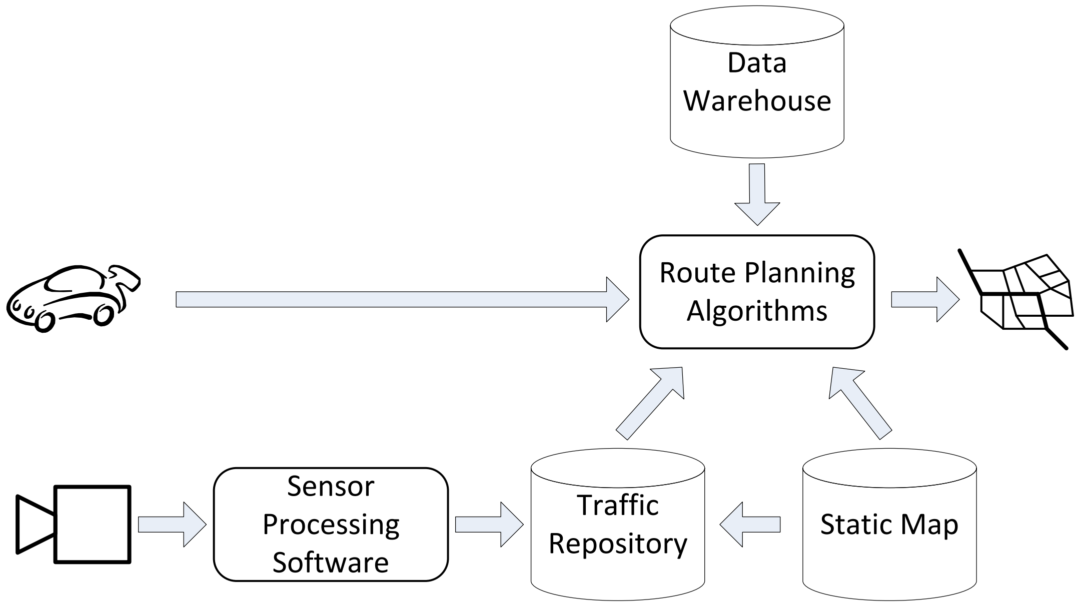
- Insigma Project and Route Planning Subsystem
- Dynamic map (and related ontologies)
- Route planning algorithms (and related ontologies)
- Architecture of the system integrating multiple algorithms
- Rules for algorithm selection

# INSIGMA Project

**Intelligent System for Global Monitoring Detection and Identification of Threats (INSIGMA):**

1. Monitoring and identification of vehicles and people,
2. Dynamic monitoring and identification of threats within the traffic,
3. **Analysis of traffic and optimizing routes for external users,**
4. Identification of suspicious people and threats,
5. Discovery of data and multimedia with the watermarking technology.

# Route Planning Subsystem – operational concept




- Road network – stored in Static Map
- Various types of sensors delivering current traffic information to the Trafic Repository: cameras and video image processors, inductive loops, acoustic arrays, mobile GPS trackers, etc.
- Data Warehouse – providing historical data,
- Various types of users: ordinary and priviledged: Police, Fire Brigades, Medical Services, Logistic Companies

# Route planning problems

**Basic problem (shortest path)**

- Weighted directed graph $G = (V, E, w)$
  - $V$ – set of vertices
  - $E \subset V \times V$ – set of edges
  - $w: E \rightarrow R^{+0}$ - weights of edges
- $s, t \in V$ – start and target nodes
- $P = \langle v_1, \dots, v_n \rangle$ path
- $w(P) = \sum_{i=2}^{n} w(v_{i-1}, v_i)$ - goal function
- Task: find a path $P: s = v_1, t = v_n$ minimizing $w(P)$

**Dynamic route plannig**:

- Weights replaced by time functions: $w(t)$
- Added $t_l$ - leave time and/or $t_a$ - arrival time

**Personalized route planning**

- Given set of users $U = \{u_1, \dots, u_n\}$ with various attributes
- Multiple graphs with different vertices, edges and weights $G(u) = (V(u), E(u), w(u))$
- Goal function parametrized by a user type: $w(P, u)$

# Route Planning Subsystem Assumptions for development

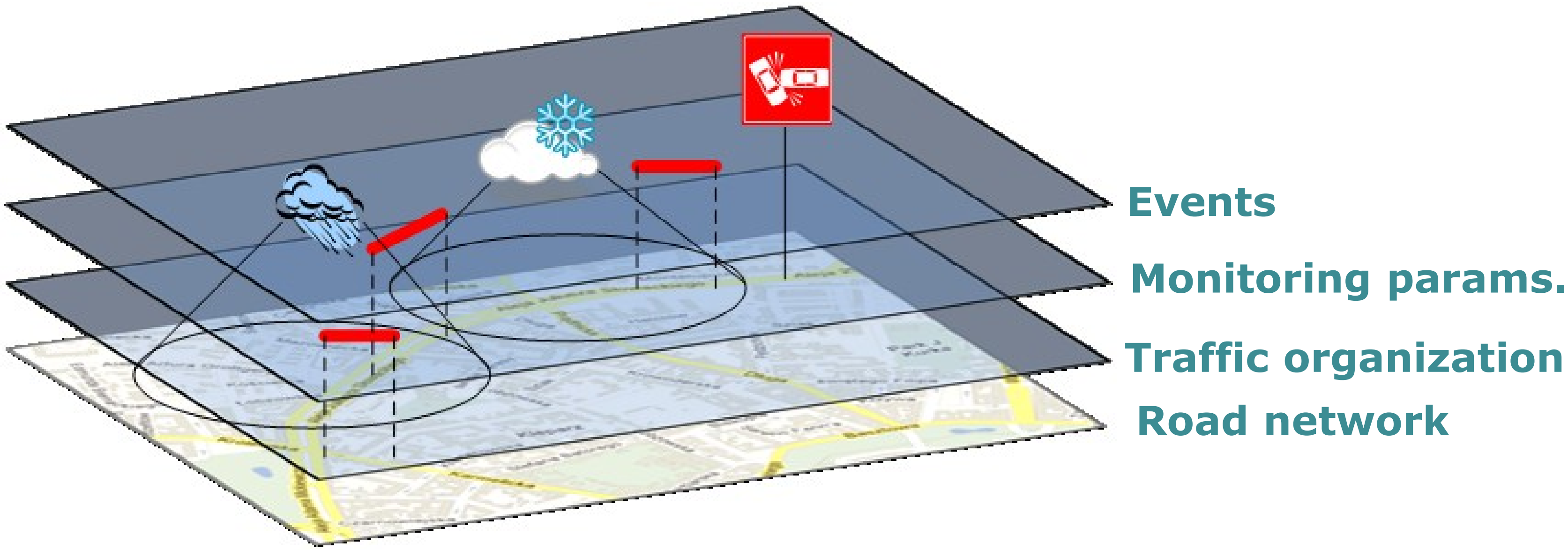
- Ontologies will be used to define all monitoring parameters stored in the Traffic Repository to enable information sharing and integration with sensors and clients. (Modifiability)
- The Route Planning Subsystem will perform dynamic and personalized planning, what favors using semantic techniques to describe user profiles, preferences and contextual information. (Customizability)
- The system will provide a framework for integrating multiple route planning algorithms and making decisions which of them should be used for a particular route planning task. (Customizability)

## Goal

Develop an open architecture of the Route Planning Subsystem supporting:

- integration with map data and dynamic information originating from various types of sensors.
- deployment and run-time integration of various algorithms (including experimental ones)

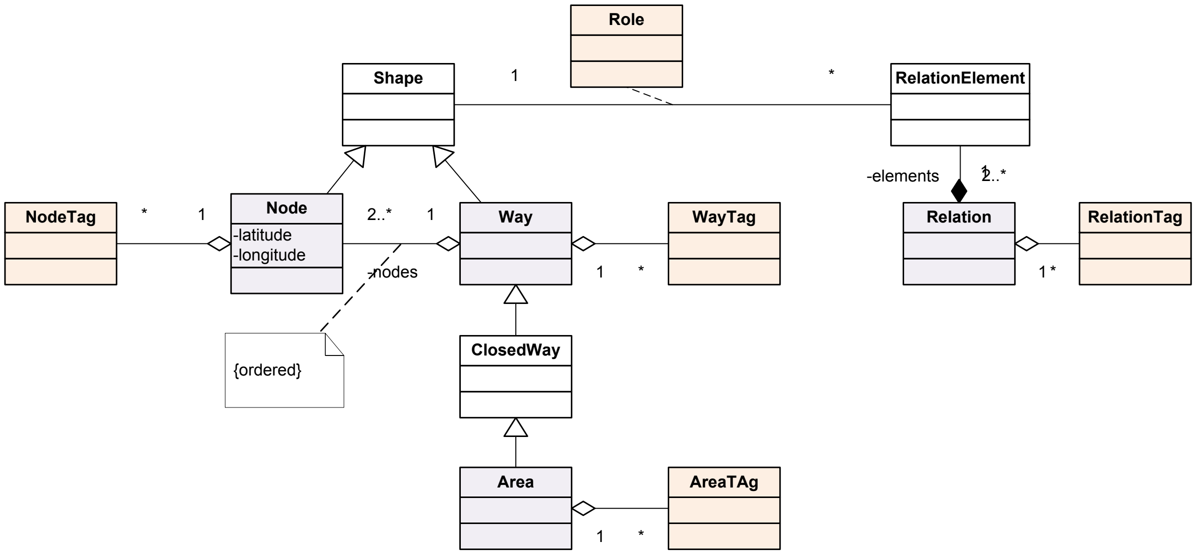
# Dynamic Map




- Consists of three relational databases wrapped by web service interfaces
  - Static map: road network and traffic organization
  - Monitoring parameters
  - Events
- Database structures generated from ontologies
- Tables contain attributes responsible for semantic mapping: URIs of classes and properties
- Interfaces support semantic queries, eg. return active instances of montitoring parameters belonging to a certain type in an indicated area.

# Dynamic map – related ontologies
# OSM ontology 1

- Information about road network and various objects originates from Open Street Map (OSM)
- Small set of graphic primitives: Node, Way, Area, Relation augmented by tags: (key,value) pairs representing several hundreds classes of map objects.



# Dynamic map – related ontologies
# OSM ontology 2

- About 660 classes of objects appearing on maps: roads, railways, water ways, amenities, emergency infrastructure, public transport, shops, tourist attractions, etc.
- Generated semi-automaticaly from HTML data
- Provides information about how these classes of map objects are represented by combinations of map primitives (Node, Way, Area) and attributed tags.
- Supports various tasks: visualization, searching for POIs (Points of Interest) and route planning.

## Dynamic map – ontologies
## Static Map Ontology

- Specifies additional data structures included into the static map: lanes, crossroads and turns
- The taxonomy of their properties (qualitative and quantitative)
  - physical characteristics (width, maximum height, turn radius, damaged surface)
  - limits or obligations imposed by the traffic organization (speed limit, forbidden turn, etc.)
  - properties originating from environment usually expressed by warning signs (wild animals, icy surface in winter, intense pedestrian traffic).

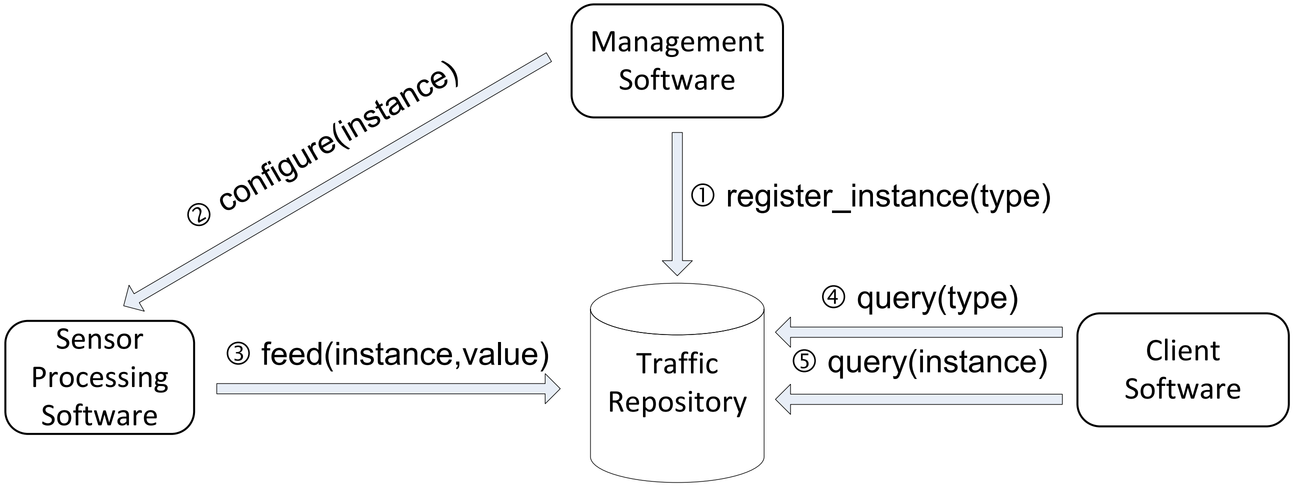
# Dynamic map – ontologies Monitoring parameters - 1

Three basic concepts:

- Monitoring parameter type
  - e.g. average speed, waiting time at the traffic lights, length of the vehicles queue, temperature, wind speed, rainfall, etc.
  - Each type is assigned with a unit, a range of values and constraints specifying map objects to which an instance of a type can be linked.
- Parameter instance
  - binds a parameter with a sensor, location where a parameter is measured, an object (node,lane, turn or area) to which the measurement applies, frequency indicating how often the data are updated and the instance state (active, suspended, failure).
- Current parameter value assigned to an instance
  - with accompanying timestamp and validity period.

# Dynamic map – ontologies Monitoring parameters - 2

- Integration with various sensor types
- Supports discovery and query services delivered by the Traffic Repository- allows client software (in particular route planning algorithms) to find monitored parameters of interesting type in an indicated area and then access their current values.
- Runtime checking of data correctness and instance/sensor state



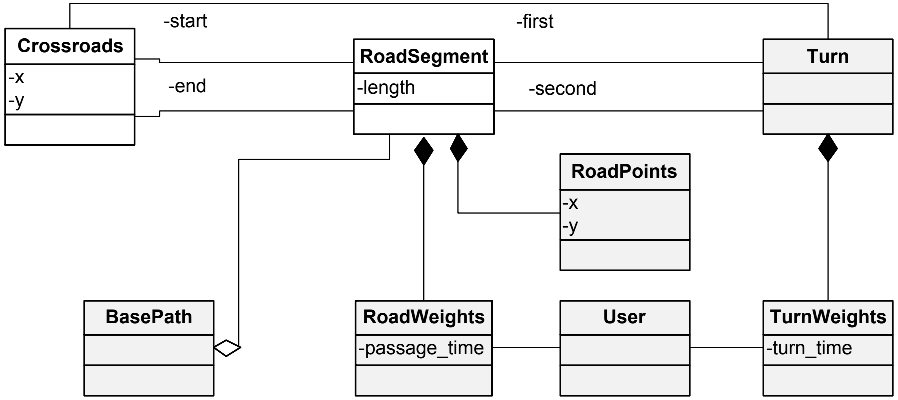
# Dynamic map – ontologies
# Events and Threats

- The ontology defines various events influencing the traffic: accidents, demonstrations, traffic jams, weather conditions, seasons.
- Events have spatiotemporal characteristic, i.e. they have an occurrence time, duration and they are attached to a certain location (point, road, area).
- The ontology also classify Threats understood as possible events that may result in loss/damage (accident) or negative influence on traffic (jam). There are casual relations between events and threats, e.g. severe weather conditions may cause accidents.

# Optimization algorithms

**Implemented algorithms** (three independent teams):

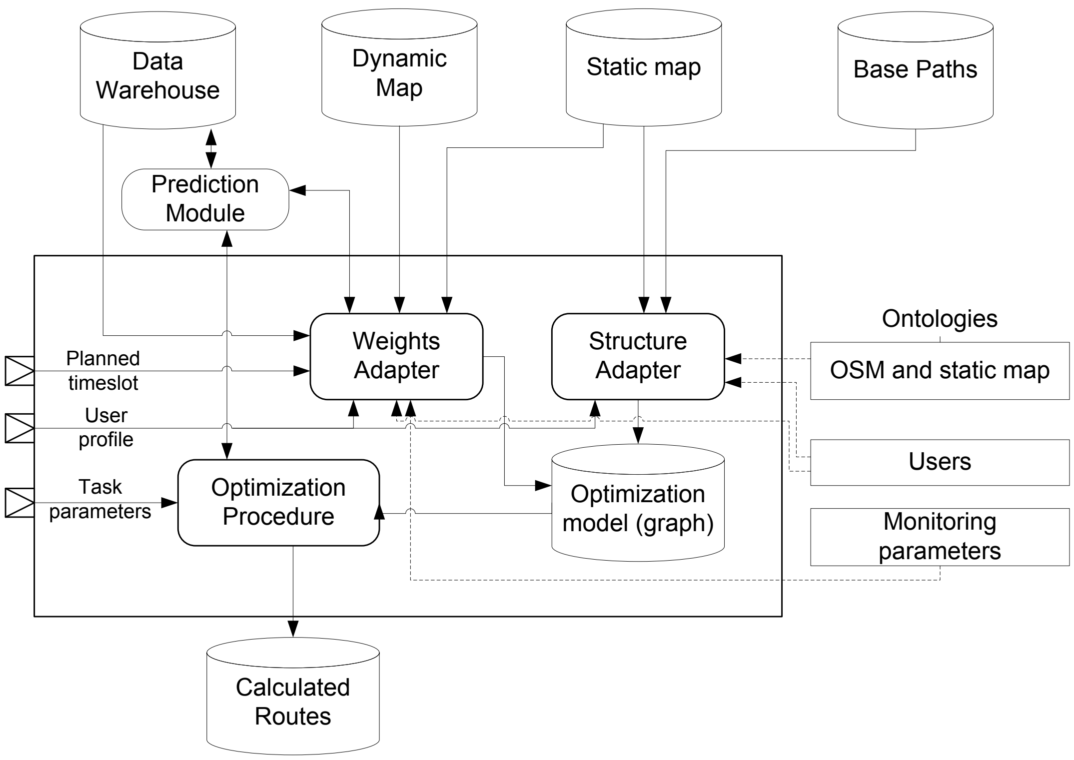
- Dijkstra
- A*
- ACO - Ant Colony Optimization
- SAMCRA
- PSO - Particle Swarm Optimization
- Tabu Search




Various graph models:

- Basic
- Modeling turns
- Geometry of road segments
- Precomputed base paths and/or hierarchical nodes and segments
- Some algorithms use adjoint graphs (edges are treated as nodes and nodes as edges)

# Algorithm implementation




- Optimization model – usually located in memory (about 170 MB for Poland)
- Structue Adapter – builds the required graph structure (based on user type); applies precomputations
- Weights adapter – calculates weights (defaults, dynamic parameters, flows of measured parameters)
- Optimization procedure – web service providing algorithm implementation

## Ontologies related to Route Planning

- Users (traffic participants)
  - Classification of vehicles (based on physical attributes: number of axles, weight and dimensions.)
  - Roles in the traffic: normal users, police, fire or medical services, etc.
- Areas
  - type of the road connections (density, road layout – regular or irregular, numerous one way roads, road width, presence of multilane roads, intense pedestrian or bicycle traffic),
  - traffic characteristics (presence of monitored parameters, average speed ranges, their variations in time, high probability of jams occurrences),
  - safety characteristics (probability of car accidents),

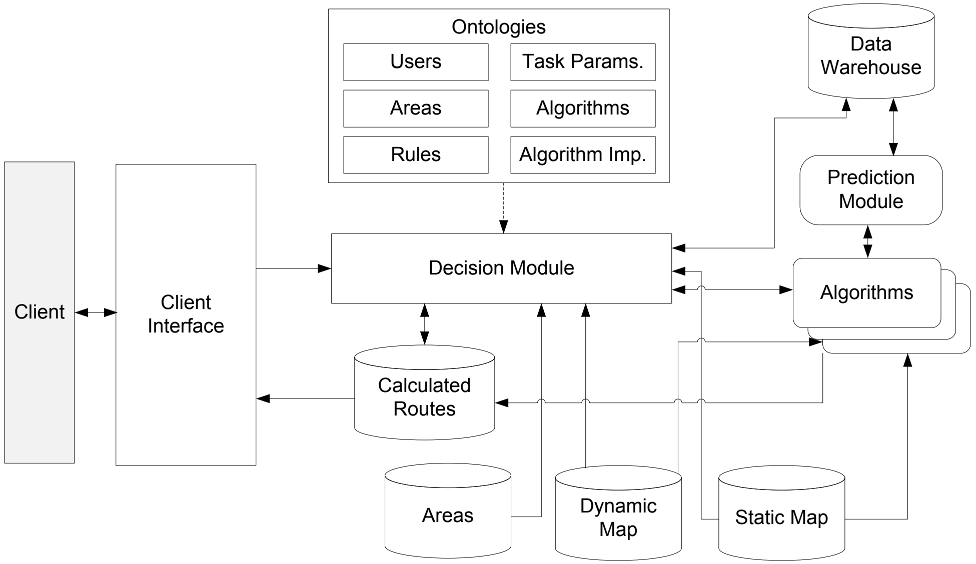
# Ontologies related to Route Planning 2

- Algorithms - classifies their types (exact, approximate, greedy), role (construction and improvement), enumerates criteria for selecting next nodes in heuristics.
- Algorithm implementations - defines addresses of web services implementing optimization procedures, describes their parameters and classifies them as instances of a particular algorithm type.
- Rules - stores rules for algorithm selection encoded in SWRL language and a narrow set of classes and properties supporting algorithm selection and configuration.

# Architecture of Route Planning Subsystem

Key assumptions:

- Web service interface
- Integrates multiple algorithms
- Decision module (broker) responsible for selection and configuration of algorithms
- Main use case: a client delivers task parameters, obtains task identifier used later for polling for routes and updating position.



# Rules

Decison module uses rules encoded in SWRL language to dispatch route planning requests to appropriate algorithm.

Elements appearing in premises and conclusions

- **Task characteristics** including defined route points (start, end, intermediary), areas to which the route points belong and their properties (street layout, density, road infrastructure, road classes, number of lanes, length of segments, intersections, traffic characteristic), distance to the base paths, travel time and distance.
- **User profile** - vehicle type, traffic role, privileges and preferences.
- **General context** (season and weather conditions).
- **Algorithms** (a class of algorithm and configuration parameters, usually flags and boolean switches).

## Rules - examples

- **SnowfallAndTruck**: In case of snow fall and a truck vehicle, use a greedy algorithm, which prefers selection of main roads.

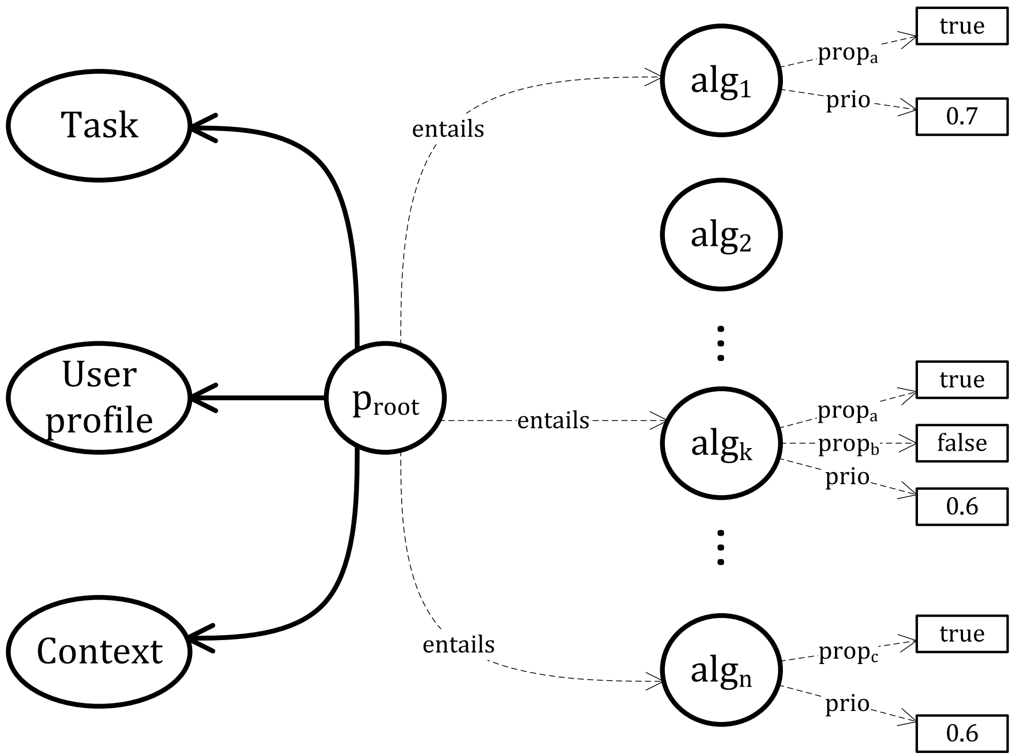
  Premise(?p),
  context(?p, ?c),Context(?c),event(?c, ?e),SnowFall(?e),
  user_profile(?p,?u),UserProfile(?u),vehicle_type(?u, ?vt),Truck(?vt)
  → entails(?p, algorithm_sf),
  GreedyAlgorithm(algorithm_sf),
  alg.preferred_main_roads(algorithm_sf, true),
  alg.priority(algorithm_sf, 0.7)

- **VelocityDisparity:** If any route point belongs to an area with observed speed disparity between roads, an algorithm should attempt to calculate routes alternative to base paths.

  Premise(?p),
  task(?p, ?t),Task(?t),task.point(?t, ?p),RoutePoint(?p),
  point.in_area(?p, ?a), Area(?a), area.velocity_disparity(?a, true)
  → entails(?p, algorithm_vd),
  alg.calculate_alternative_routes(algorithm_vd, true),
  alg.priority(algorithm_vd, 0.65)

# Evaluation of rules

1. Construct in memory a temporary Abox with $p_{root}$ belonging to the class $Premise$, link it with $Task$, $User\ profile$ and $Context$ individuals, assert relations.
2. For each rule $r_i$ add an individual describing algoritm configuration $alg_i$ and link it with $p_{root}$ by $entails$ relation.
3. Run Pellet to execute set of rules
4. Merge the list of algorithms and configurations based on priorities




The whole process takes 200-500 ms, however require preparing an ontolgy view.

## Merging results

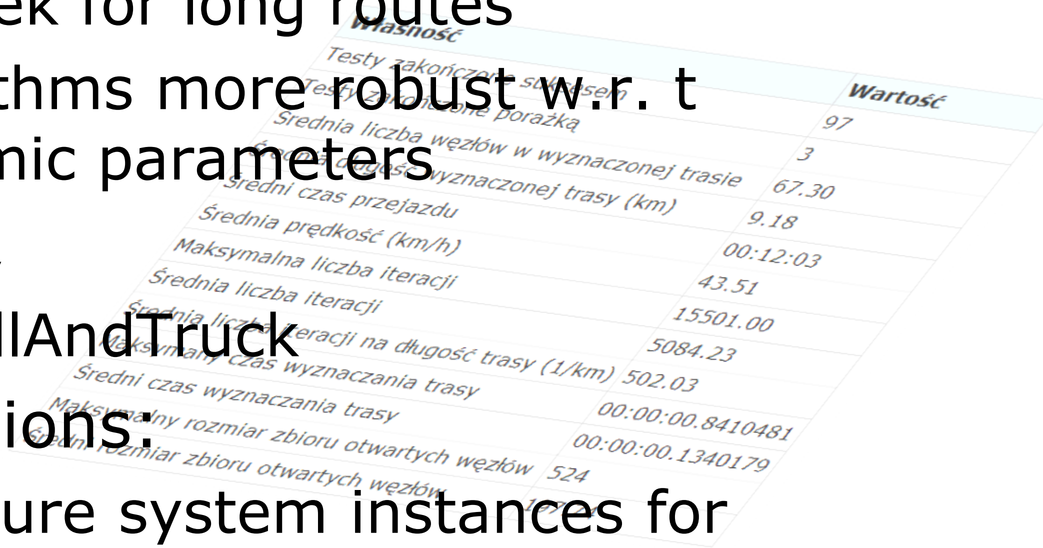
Simple heuristic procedure

- $conflict(a_1, a_2)$ relation between configurations can be estabilished while designing rules

$A_0 = \emptyset$
$\mathbf{for}\ i = 0 \ldots n\ \mathbf{do}$
$\mathbf{if}\ \forall a \in A_0 \neg conflict(a, a_i)\ \mathbf{then\ add}\ a_i\ \mathbf{to} A_0$
$\mathbf{merge\ all\ graphs\ from} A_0$

# Sources of rules

- Randomly generated thousands of test cases:
  - exact A* algorithm good in urban environment, week for long routes
  - population algorithms more robust w.r. t changes in dynamic parameters
- Expert knowledge,
  - e.g. rule SnowFallAndTruck
- Architectural decisions:
  - e.g. separate secure system instances for police or fire brigades including extended model of road networks.



# Conclusions and outlook

- Ontologies used at **development time** to formalize domain models and generate database structures
- At **run time:**
  - support integration with Dynamic Map (sensors and clients)
  - decision making - dispatching route planning request to appropriate algorithm implementation

**Future works**

- Improve coverage and reliability of dynamic weights:
  - New sensor types, in particular GPS trackers
  - Models supporting estimation of traffic based on measured parameters
- Learning priorities of rules (in general by launching all algorithm instances for a number of tasks and comparing results)
- Automatic clustering of roads into areas