Multiagent Planning

from PAH perspective

MA Planning Concept

Single-agent reasoning

MA Planning Concept

Single-agent reasoning

- planning
- control
- diagnostics
- verification
- problem solving
- statistical learning

MA Planning Concept

Single-agent reasoning

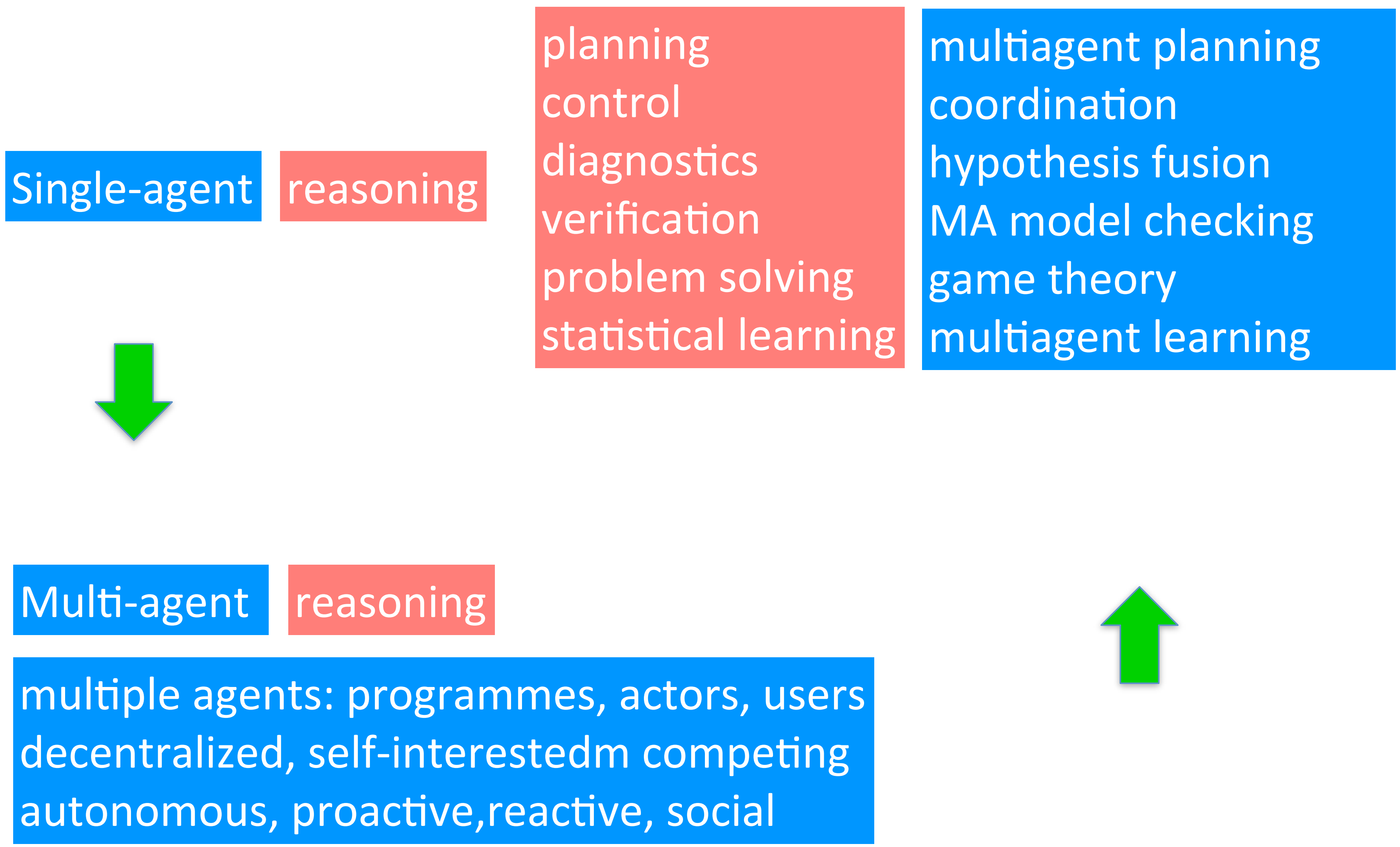
planning
control
diagnostics
verification
problem solving
statistical learning

↓

Multi-agent reasoning

multiple agents: programmes, actors, users
decentralized, self-interestedm competing
autonomous, proactive,reactive, social

MA Planning Concept



Agents briefly (one slide only)

- *multi-agent system* is a decentralized multi-actor (software) system, often geographically distributed whose behavior is defined and implemented by means of complex, peer-to-peer interaction among autonomous, rational and deliberative entities.
- *autonomous agent* is a special kind of a intelligent software program that is capable of highly autonomous rational action, aimed at achieving the private objective of the agent – can exists on its own but often is a component of a multi-agent system – agent is autonomous, reactive, proactive and social
- *agent researchers* study problems of integration, communication, reasoning and knowledge representation, competition (games) and cooperation (robotics), agent oriented software engineering, …

Agents briefly (one slide only)

- *agent technology* is software technology supporting the development of the autonomous agents and multi-agent systems *agent-based computing* is a special research domain, subfield of computer science and artificial intelligence that studies the concepts of autonomous agents
- *multi-agent application* is a software system, functionality of which is given by interaction among autonomous software/hardware/human components.

  But also a monolithic software application that is autonomously operating within a community of autonomously acting software applications, hardware systems or human individuals.

Opportunities for MAS Deployment

Agent-based computing have been used:

1. **Design paradigm** – the concept of decentralized, interacting, socially aware, autonomous entities as underlying software paradigm (often deployed only in parts, where it suits the application)
2. **Source of technologies** – algorithms, models, techniques architectures, protocols but also software packages that facilitate development of multi-agent systems
3. **Simulation concept** – a specialized software technology that allows simulation of natural multi-agent systems, based on (1) and (2).

Opportunities for MAS Deployment

Agent-based computing have been used:

1. **Design paradigm** – the concept of decentralized, interacting, socially aware, autonomous entities as underlying software paradigm (often deployed only in parts, where it suits the application)

↓

- *Agent Oriented Software Engineering* – provide designers and developers with a way of structuring an application around autonomous, communicative elements, and lead to the construction of software tools and infrastructures to support this metaphor.

Opportunities for MAS Deployment

Agent-based computing have been used:

1. **Design paradigm** – the concept of decentralized, interacting, socially
2. **Source of technologies** – algorithms, models, techniques architectures, protocols but also software packages that facilitate development of multi-agent systems

- *Multi-Agent Techniques* – provide a selection of specific computational techniques and algorithms for dealing with collective of computational processes and complexity of interactions in dynamic and open environments.
- *Autonomy-Oriented Techniques* – provide set of artificial intelligence techniques supporting autonomous decision making of intelligent systems and methods of adjusting their decision making autonomy.

Opportunities for MAS Deployment

Agent-based computing have been used:

1. **Design paradigm** – the concept of decentralized, interacting, socially
2. **Source of technologies** – algorithms, models, techniques architectures,
3. **Simulation concept** – a specialized software technology that allows simulation of natural multi-agent systems, based on (1) and (2).

- *Multi-Agent Simulation* – provide expressive models for representing complex and dynamic real-world environments, with the emphasis on capturing the interaction related properties of such systems.

Multiagent planning

- **Multiagent planning has been viewed as either**
  1. planning for activities and resources allocated among distributed agents,
  2. distributed (parallel) computation aimed at plan construction or
  3. plan merging/coordination activity.
- **multiagent planning approaches according to:**
  - Durfee
  - Domshlack
  - Sislak

Multiagent planning (Durfee)

- The classical work of Durfee divides the planning process into:
  1. task decomposition
  2. subtask allocation
  3. conflict detection
  4. individual planning
  5. plan merging

Multiagent planning (Durfee)

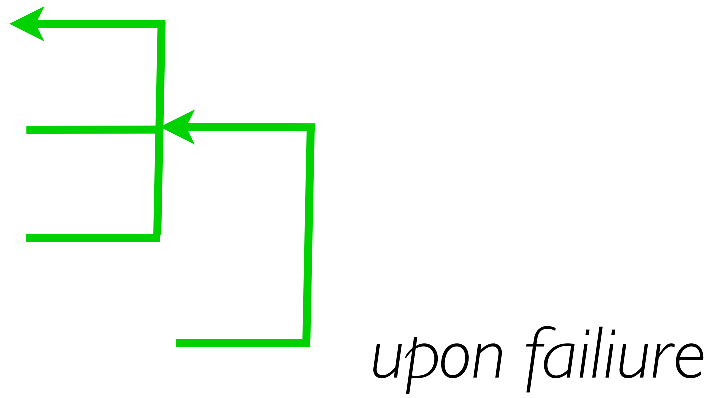
- **The classical work of Durfee divides the planning process into:**
  1. task decomposition
  2. subtask allocation
  3. conflict detection
  4. individual planning
  5. plan merging

negotiation, social knowledge modeling

- *Negotiation:* multiagent methods for agreeing on a joint, collective decision that involves autonomous, self-interested agents:
  - combinatorial auctions, voting protocols, ...
- *Social knowledge* (SK): knowledge structures maintained by the agents that models their provided capability (SK0 → SK1 → SK2 ...)

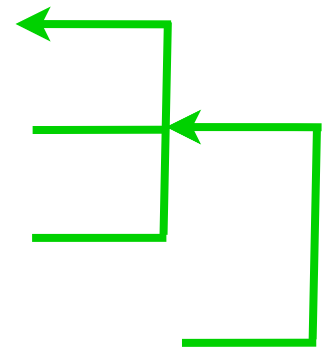
Multiagent planning (Durfee)

- **The classical work of Durfee divides the planning process into:**
  1. task decomposition
  2. subtask allocation
  3. conflict detection
  4. individual planning
  5. plan merging



Multiagent planning (Durfee)

- **The classical work of Durfee divides the planning process into:**
  1. task decomposition
  2. subtask allocation
  3. conflict detection
  4. individual planning
  5. plan merging



upon failiure

  - *Multiagent planning coordination:* extension of partial order, causal link (POCL) definition of a plan.They provide formal definition of the multi-agent parallel POCL plan, where they introduce parallel step thread flaw and plan merge step flaw. Based on this they have designed a multi-agent plan coordination algorithm that is working with the space of complete plans of the individual agents.
  - The algorithm is based on branch-and-bounds search.

Multiagent planning (Durfee)

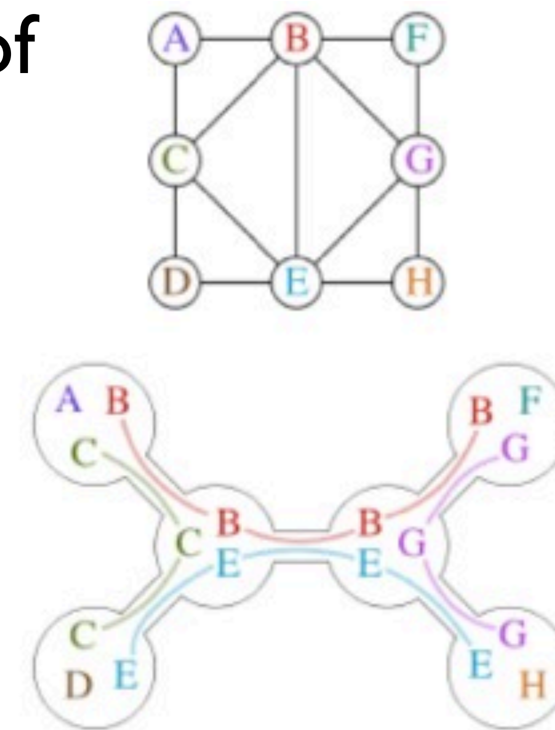
- The classical work of Durfee divides the planning process into:
  1. task decomposition
  2. subtask allocation
  3. conflict detection
  4. individual planning
  5. plan merging

Multiagent planning (Domshlack)

- Tries to minimize the concept of centrality in MA planning
- Assumes that there is no 1. - 2. phase of Durfee MA planning
  - all agents can take initiative
- Assumes that the community of agents is *loosely coupled*

Multiagent planning (Domshlack)

- **Approach: graphical representation for definition of**
  - a *system coupling level* (based on interactions and *problem coupling level* (minmax number of times a single agent needssome other agent to do something for it)

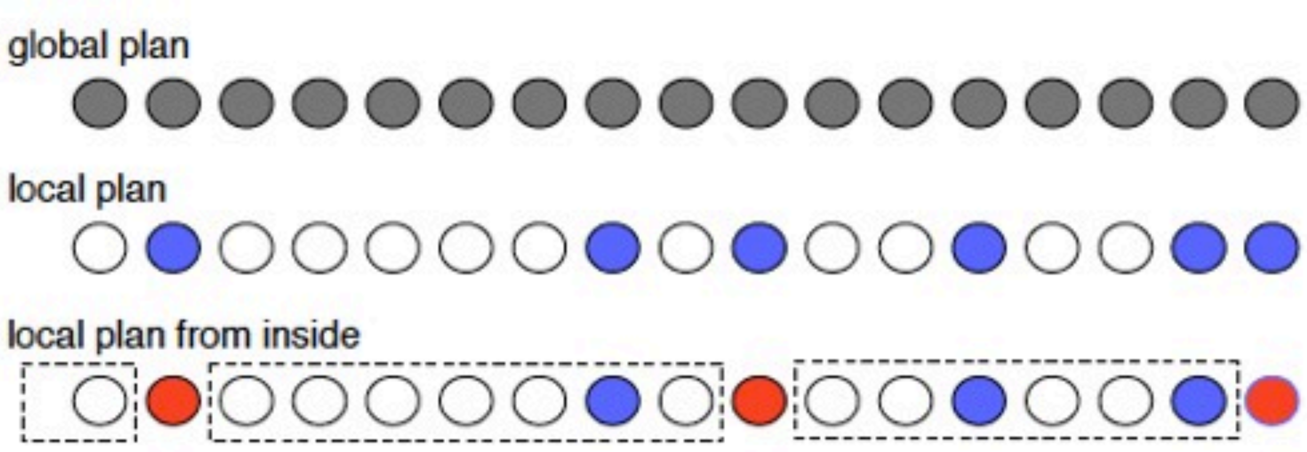


- **Algorithm:**
  1. Fix the agents' commitments to each other
  2. Let each agent independently plan *in-between* commitments
  3. Use iterative deepening to extend the number of commitments if needed

Multiagent planning (Domshlack)

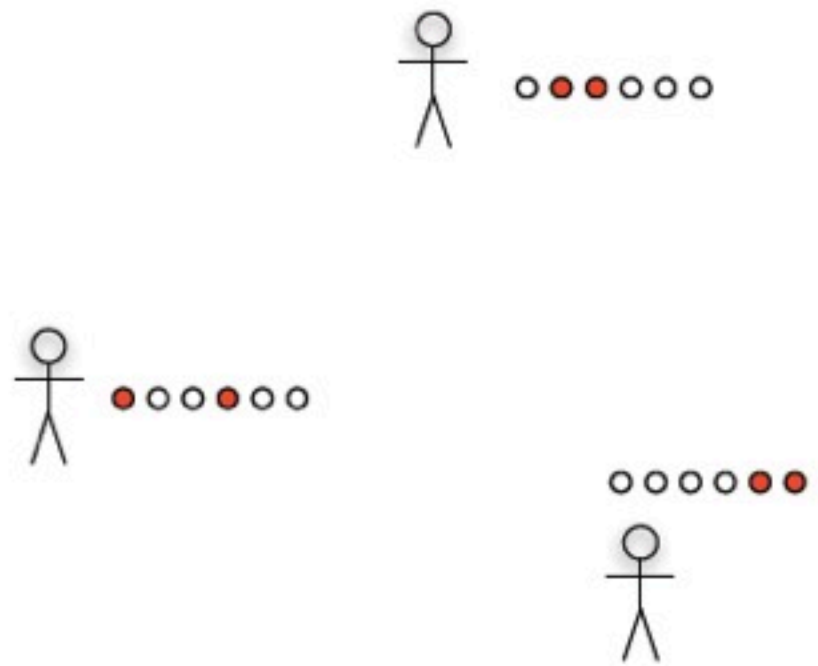
- Algorithm:
  1. Fix the agents' commitments to each other
  2. Let each agent independently plan *in-between* commitments
  3. Use iterative deepening to extend the number of commitments if needed

- Private vs. Non-Private
  - Private action affect ect and depend only on that agent
  - Non-private action is a coordination point



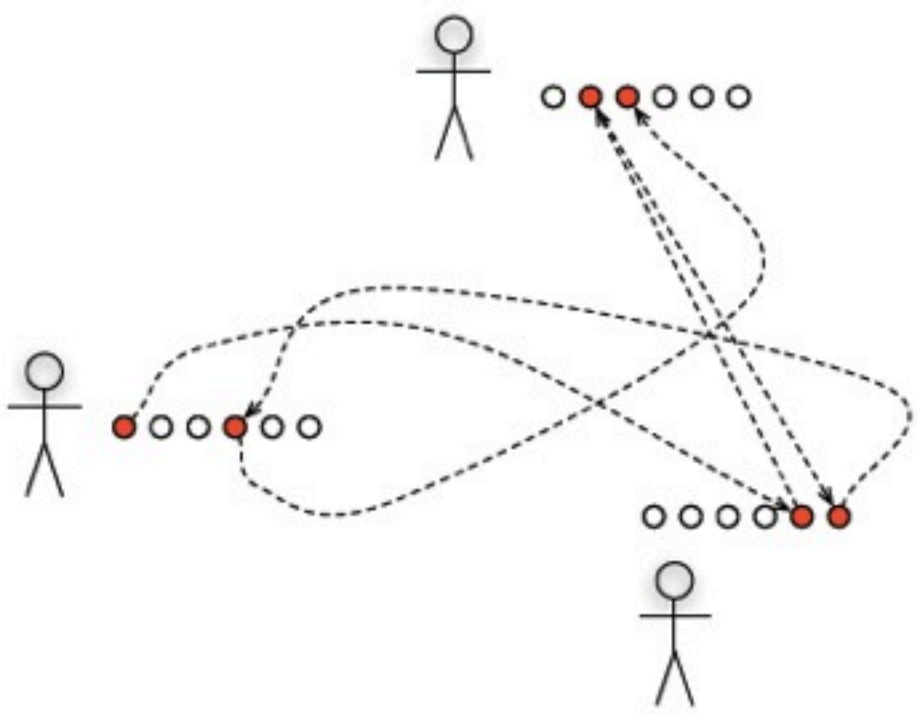
Multiagent planning (Domshlack)

- Algorithm:
  1. Find a good choice of coordination points
  2. Solve *k* local planning problems over the private actions of the agents only



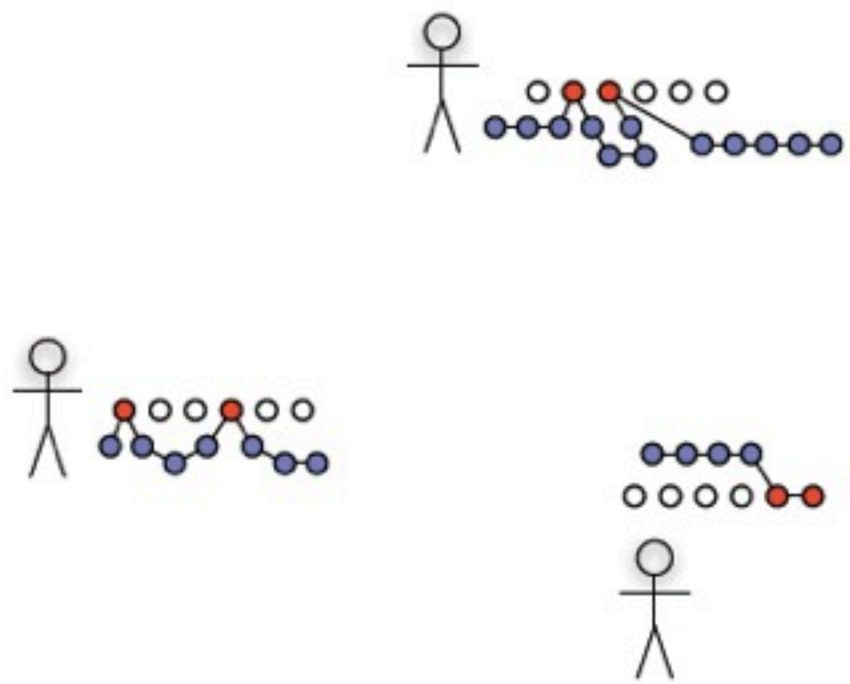
Multiagent planning (Domshlack)

- **Algorithm:**
  1. Find a good choice of coordination points
  2. Solve *k* local planning problems over the private actions of the agents only



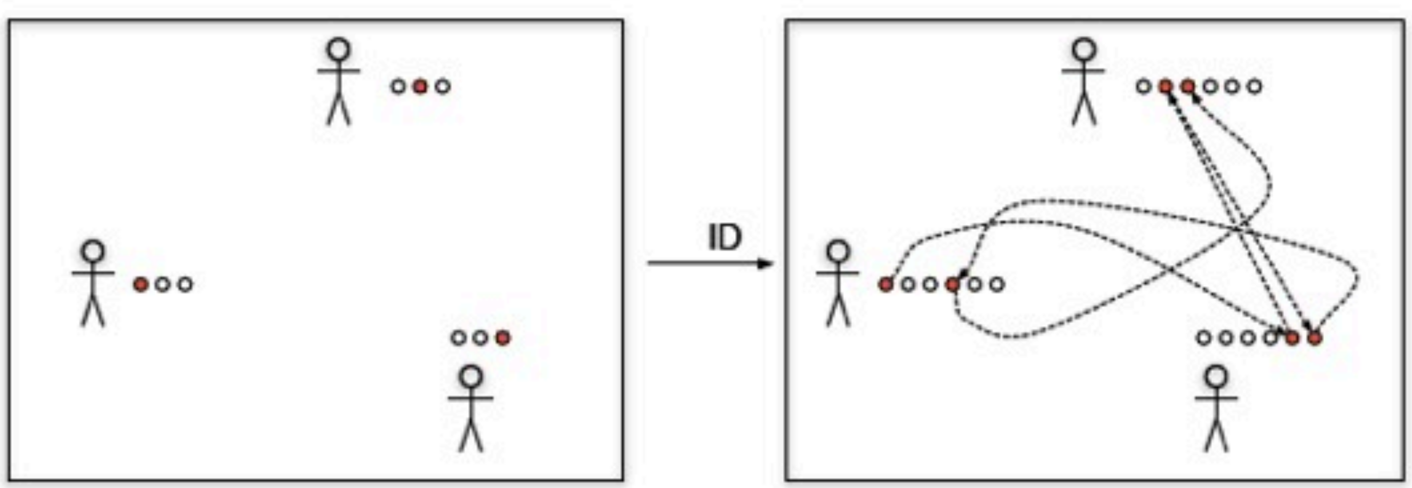
Multiagent planning (Domshlack)

- **Algorithm:**
  1. Find a good choice of coordination points
  2. Solve *k* local planning problems over the private actions of the agents only



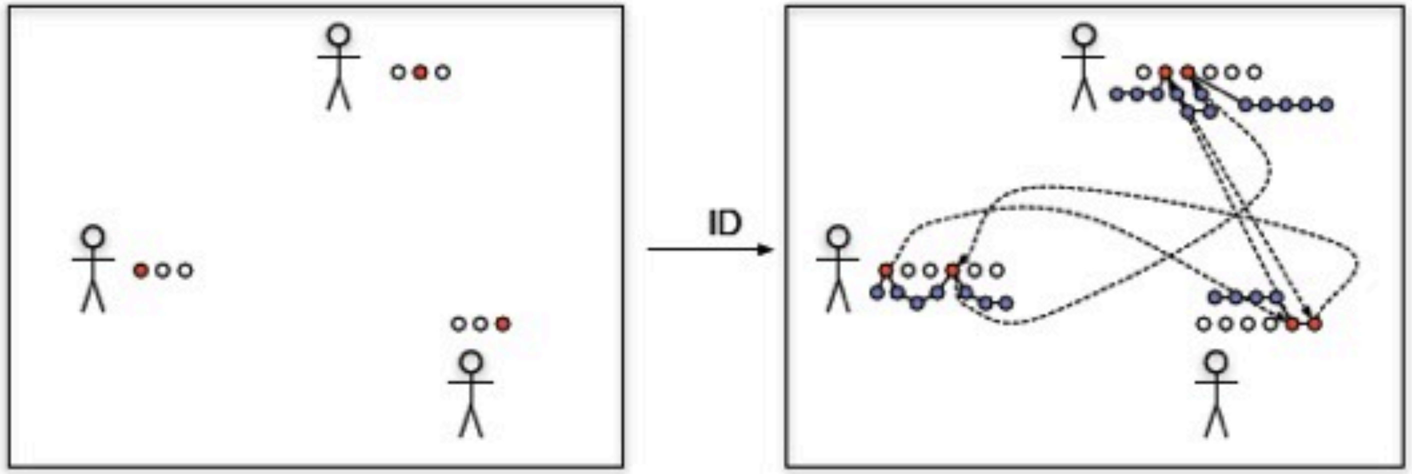
Multiagent planning (Domshlack)

- **Algorithm:**
  1. Find a good choice of coordination points
     - *Iterative deepening on # of coord-points per agent: For each choice of Define a CSP whose solutions are consistent assignments to the coordination points*
  2. Solve *k* local planning problems over the private actions of the agents only



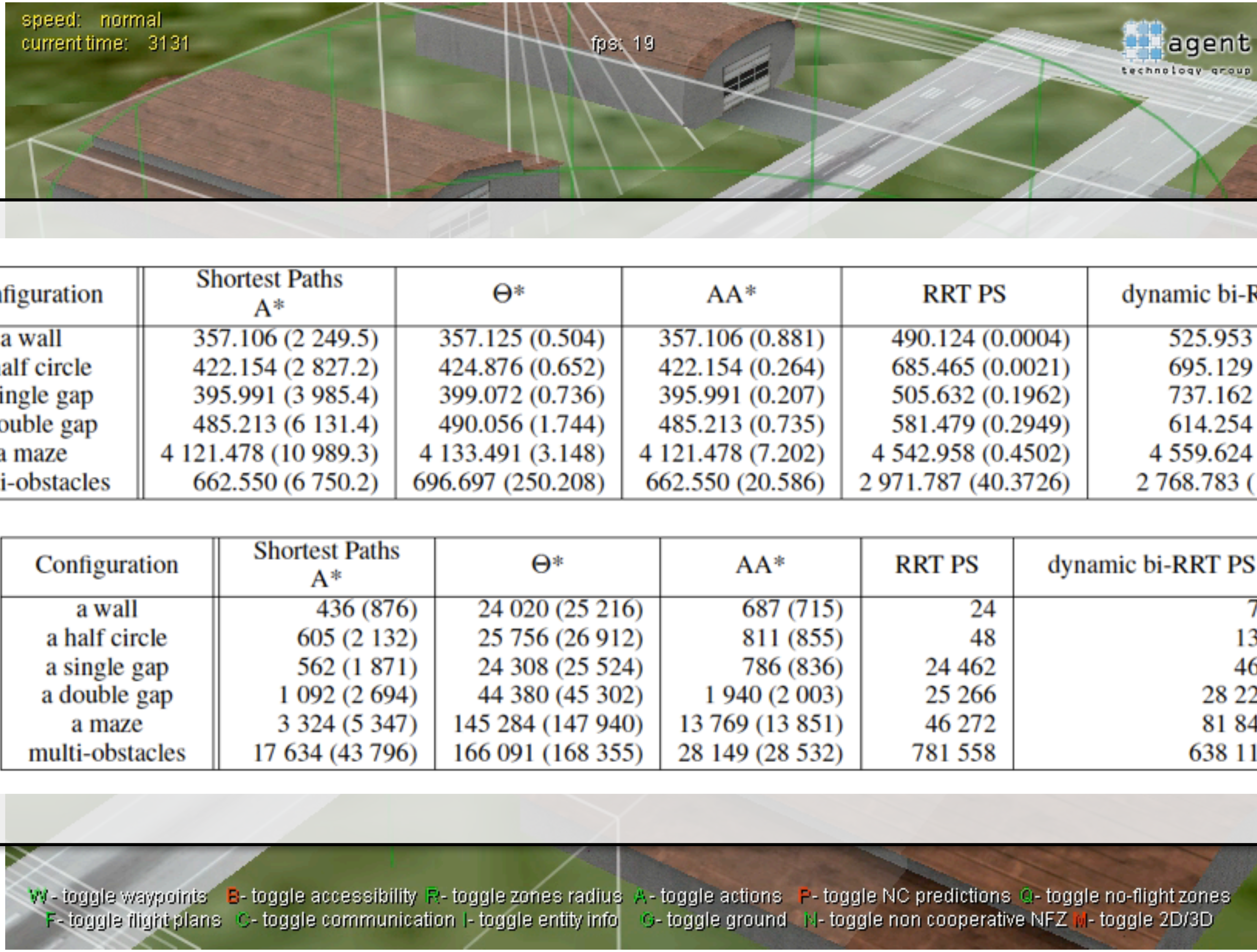
Multiagent planning (Domshlack)

- **Algorithm:**
  1. Find a good choice of coordination points
     - *Iterative deepening on # of coord-points per agent: For each choice of Define a CSP whose solutions are consistent assignments to the coordination points*
  2. Solve *k* local planning problems over the private actions of the agents only
     - *purely independent phase can be reduced to regular strips planning*



Multiagent planning (Sislak)

- Objective:
  - *Find a collective trajectory of multiple UAS that occupy same airspace and coordinate (by sense-and-avoid) their plans*



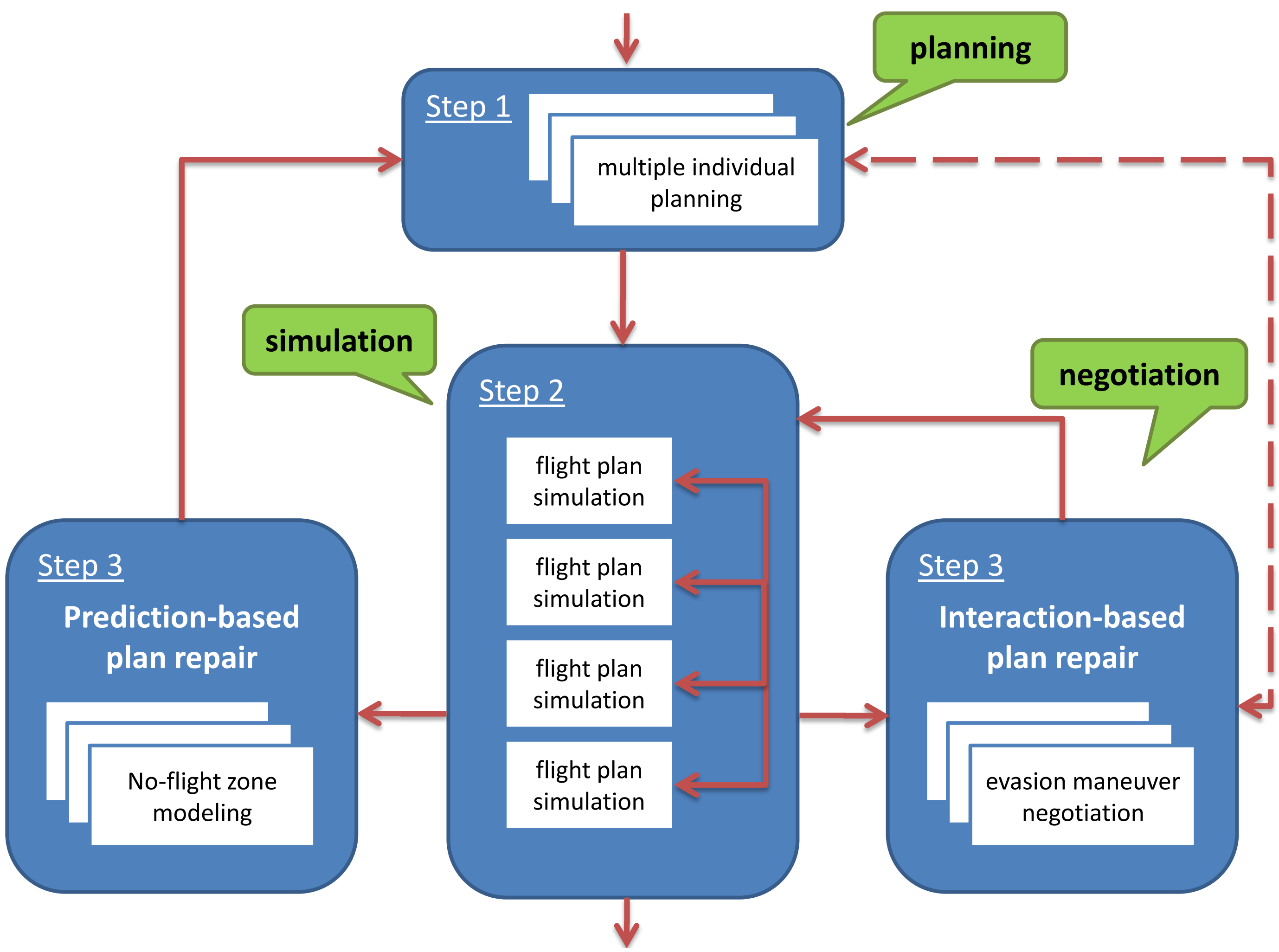
Configuration	Shortest Paths A*	Θ*	AA*	RRT PS	dynamic bi-RRT PS
a wall	357.106 (2 249.5)	357.125 (0.504)	357.106 (0.881)	490.124 (0.0004)	525.953 (0.0001)
a half circle	422.154 (2 827.2)	424.876 (0.652)	422.154 (0.264)	685.465 (0.0021)	695.129 (0.0001)
a single gap	395.991 (3 985.4)	399.072 (0.736)	395.991 (0.207)	505.632 (0.1962)	737.162 (0.0006)
a double gap	485.213 (6 131.4)	490.056 (1.744)	485.213 (0.735)	581.479 (0.2949)	614.254 (0.2173)
a maze	4 121.478 (10 989.3)	4 133.491 (3.148)	4 121.478 (7.202)	4 542.958 (0.4502)	4 559.624 (0.0188)
multi-obstacles	662.550 (6 750.2)	696.697 (250.208)	662.550 (20.586)	2 971.787 (40.3726)	2 768.783 (17.7803)

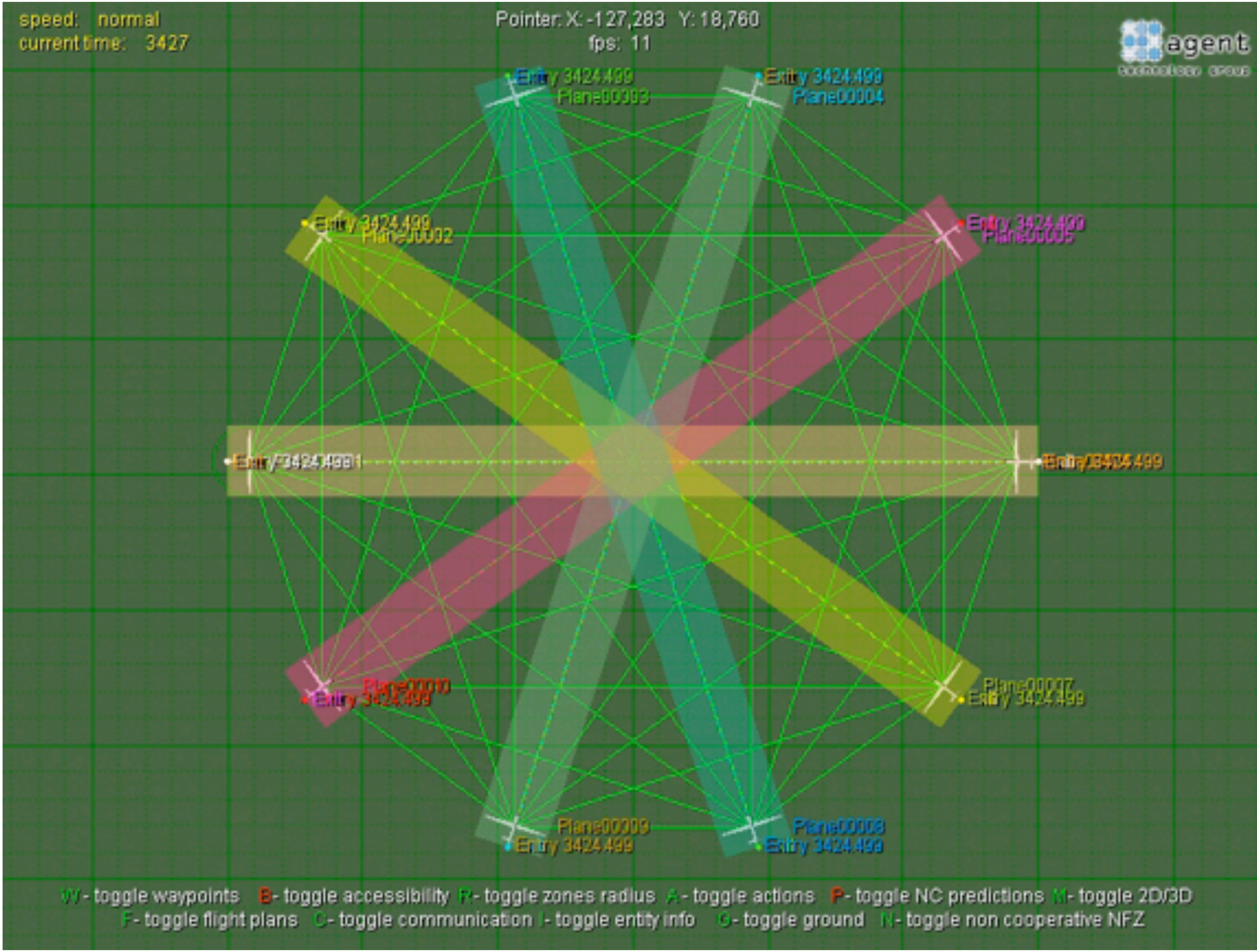
Configuration	Shortest Paths A*	Θ*	AA*	RRT PS	dynamic bi-RRT PS
a wall	436 (876)	24 020 (25 216)	687 (715)	24	74
a half circle	605 (2 132)	25 756 (26 912)	811 (855)	48	131
a single gap	562 (1 871)	24 308 (25 524)	786 (836)	24 462	463
a double gap	1 092 (2 694)	44 380 (45 302)	1 940 (2 003)	25 266	28 228
a maze	3 324 (5 347)	145 284 (147 940)	13 769 (13 851)	46 272	81 843
multi-obstacles	17 634 (43 796)	166 091 (168 355)	28 149 (28 532)	781 558	638 117

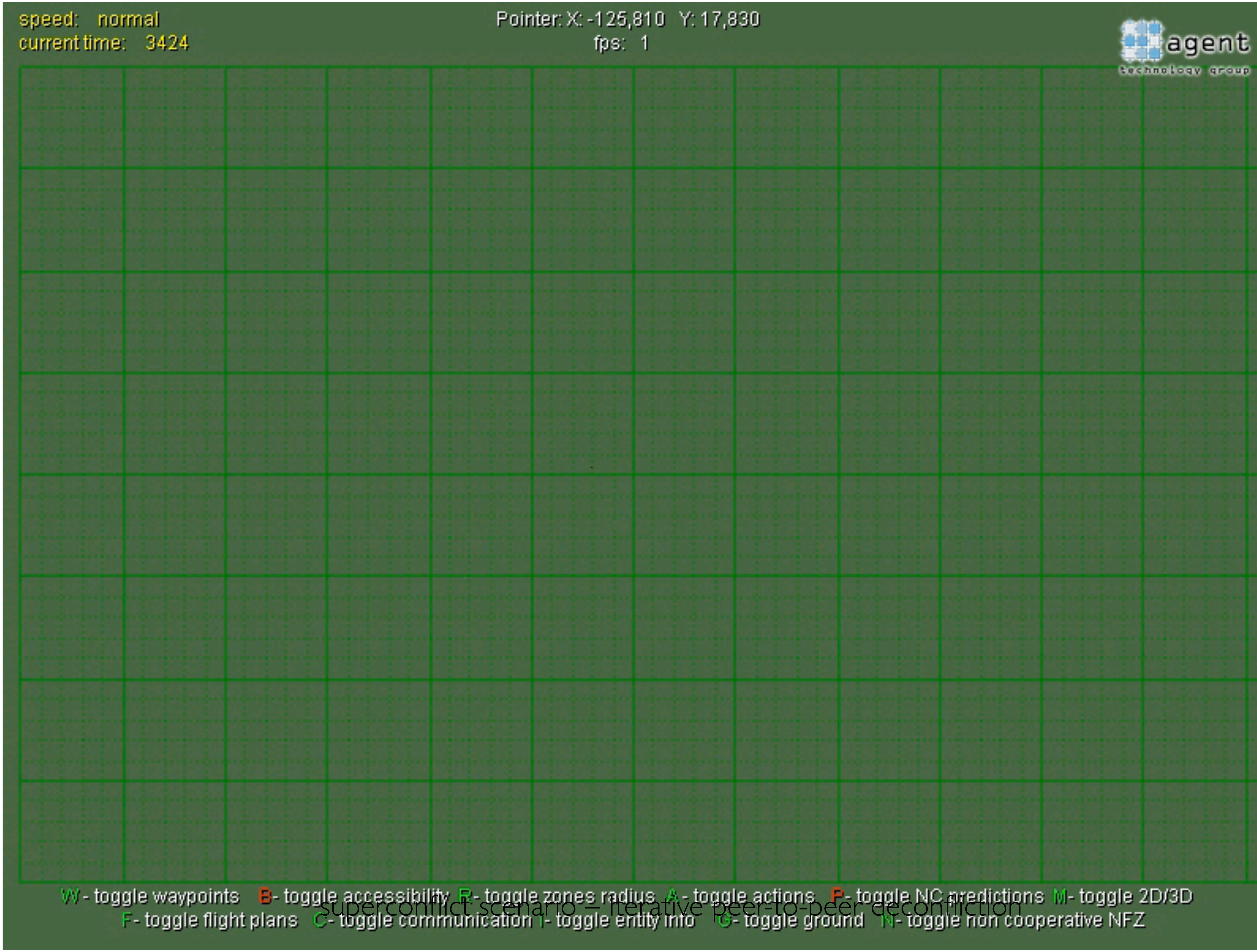
AgentFly trajectory planning

Multiagent planning (Sislak)

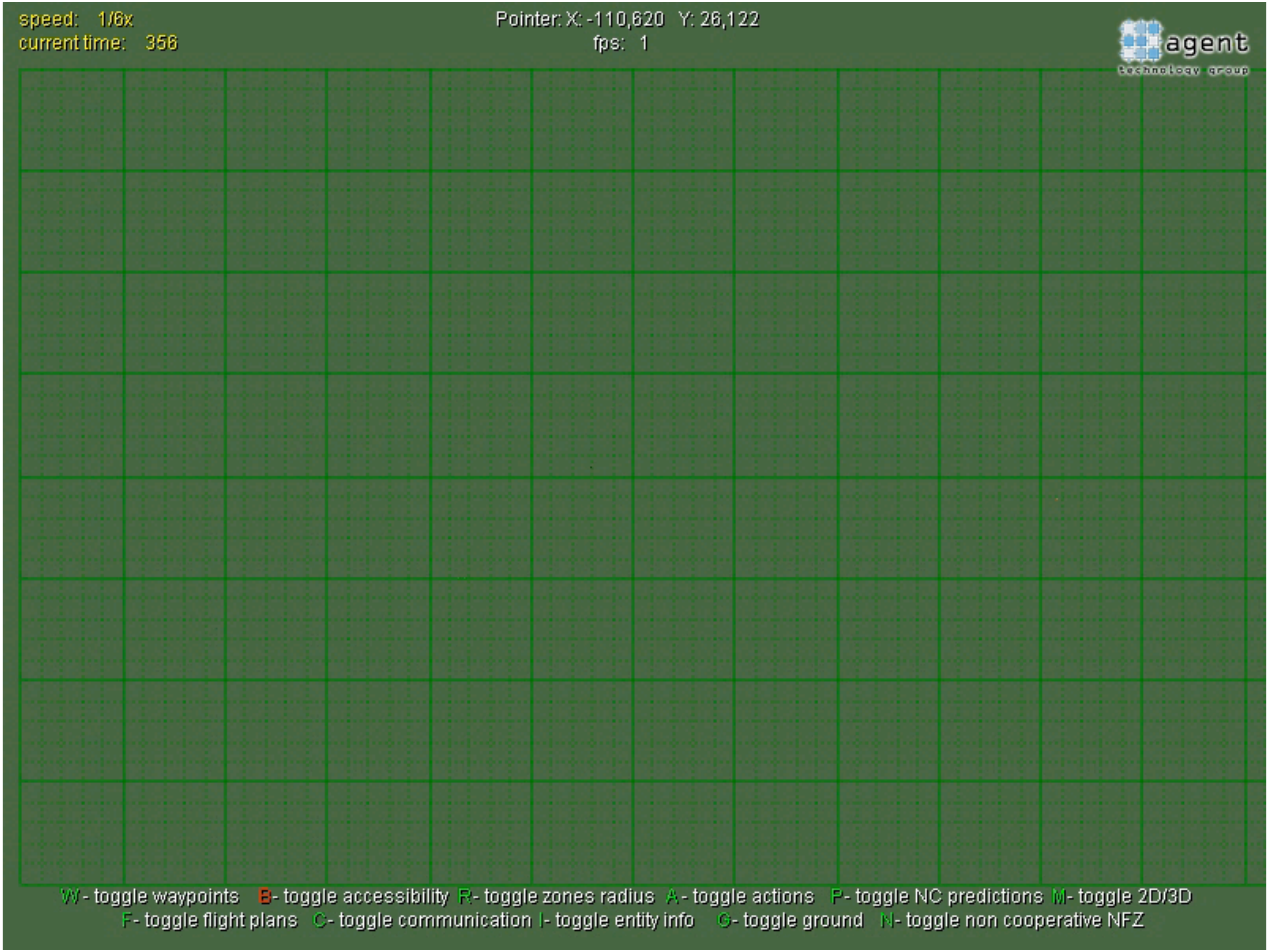
- Objective:
  - *Find a collective trajectory of multiple UAS that occupy same airspace and coordinate (by sense-and-avoid) their plans*



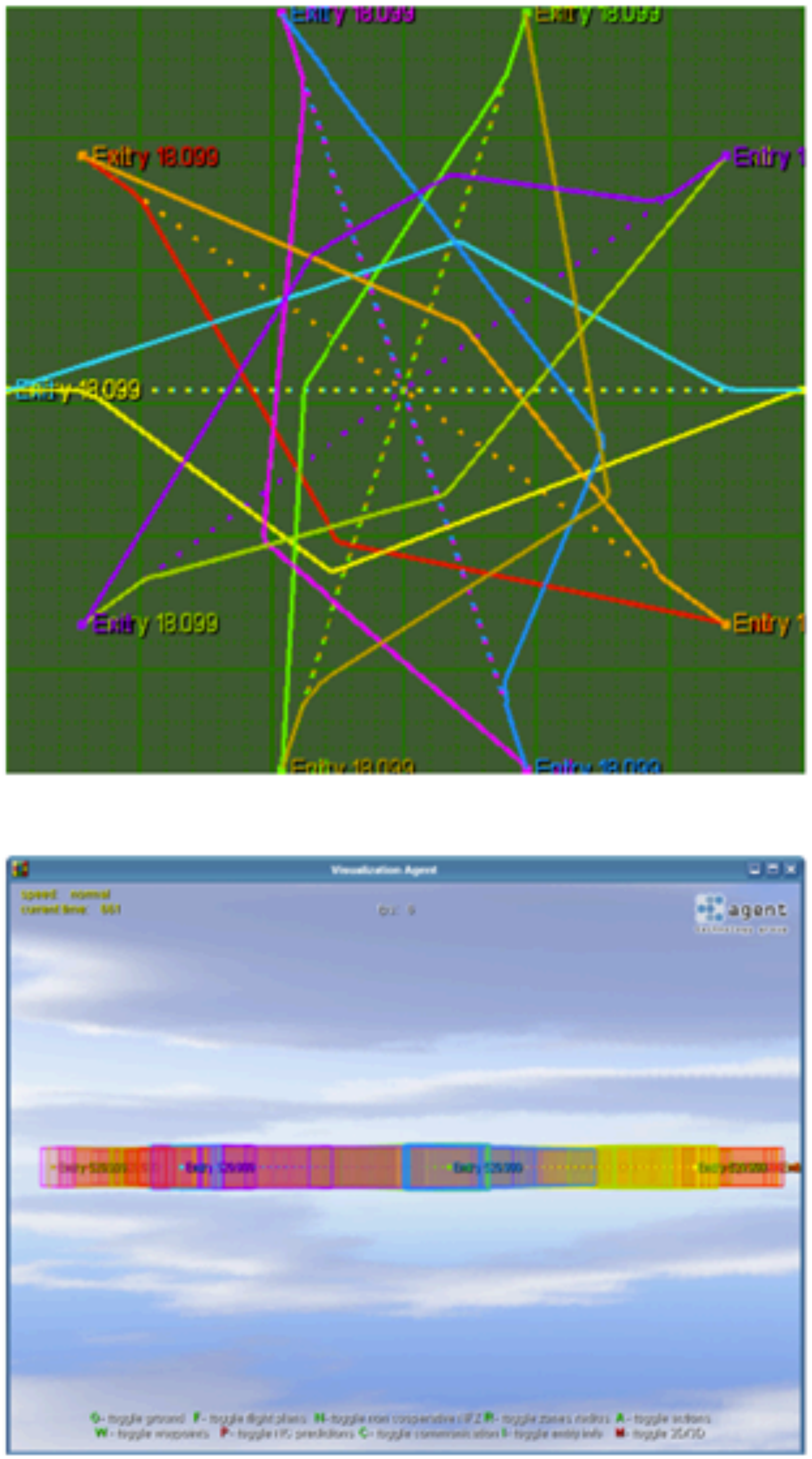

speed: normal
current time: 3427
Pointer: X: -127,283 Y: 18,760
fps: 11
agent
Entry 3424.499
Plane00003
Entry 3424.499
Plane00004
Entry 3424.499
Entry 3424.499
Plane00007
Entry 3424.499
Plane00009
Entry 3424.499
Plane00008
Entry 3424.499
W - toggle waypoints
B - toggle accessibility
R - toggle zones radius
A - toggle actions
P - toggle NC predictions
M - toggle 2D/3D
F - toggle flight plans
C - toggle communication
I - toggle entity info
G - toggle ground
N - toggle non cooperative NFZ



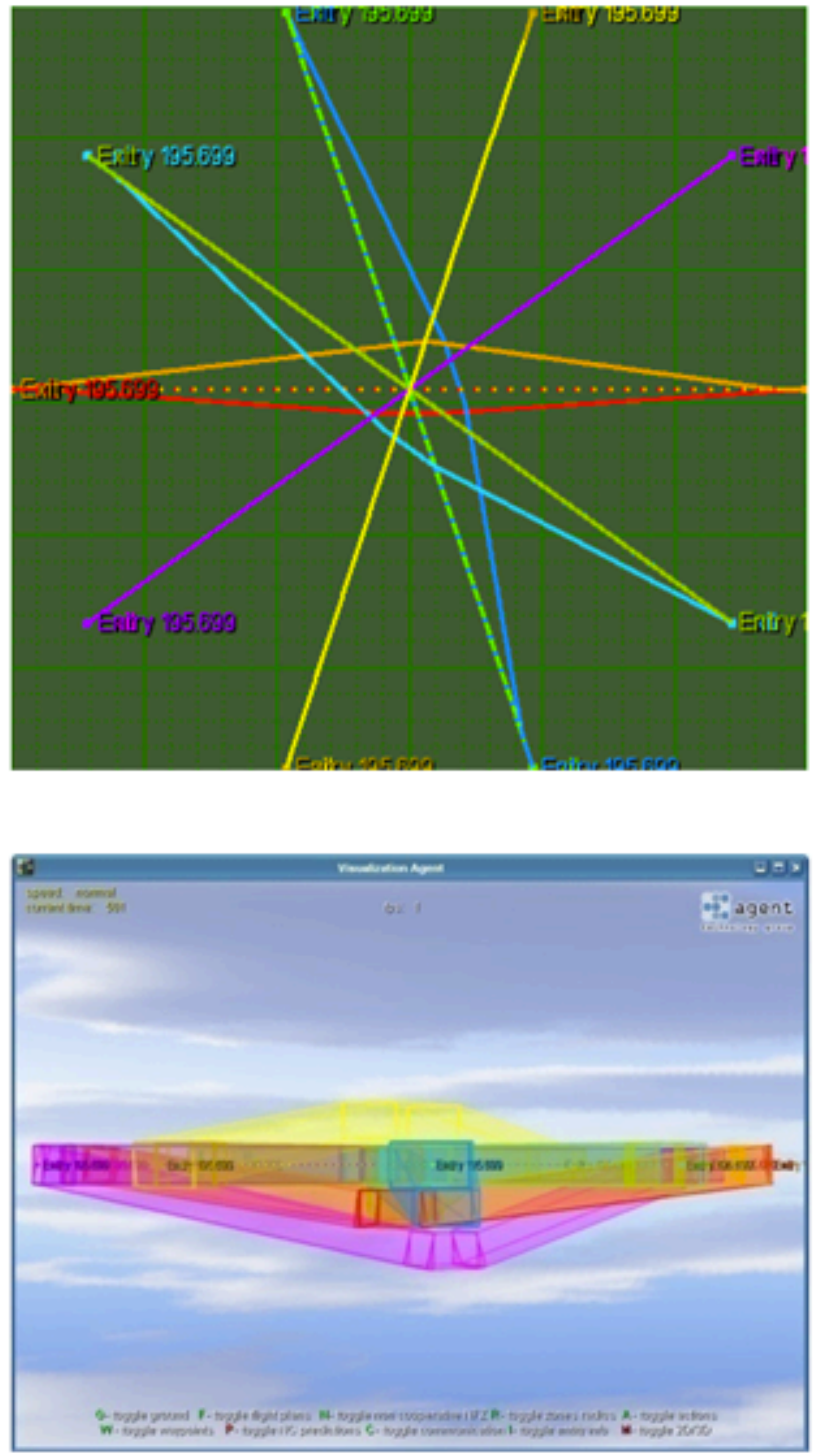
superconflict scenario – iterative peer-to-peer deconfliction


speed: 1/6x
current time: 356
Pointer: X: -110,620 Y: 26,122
fps: 1
agent
technology group
W - toggle waypoints B - toggle accessibility R - toggle zones radius A - toggle actions P - toggle NC predictions M - toggle 2D/3D
F - toggle flight plans C - toggle communication I - toggle entity info G - toggle ground N - toggle non cooperative NFZ

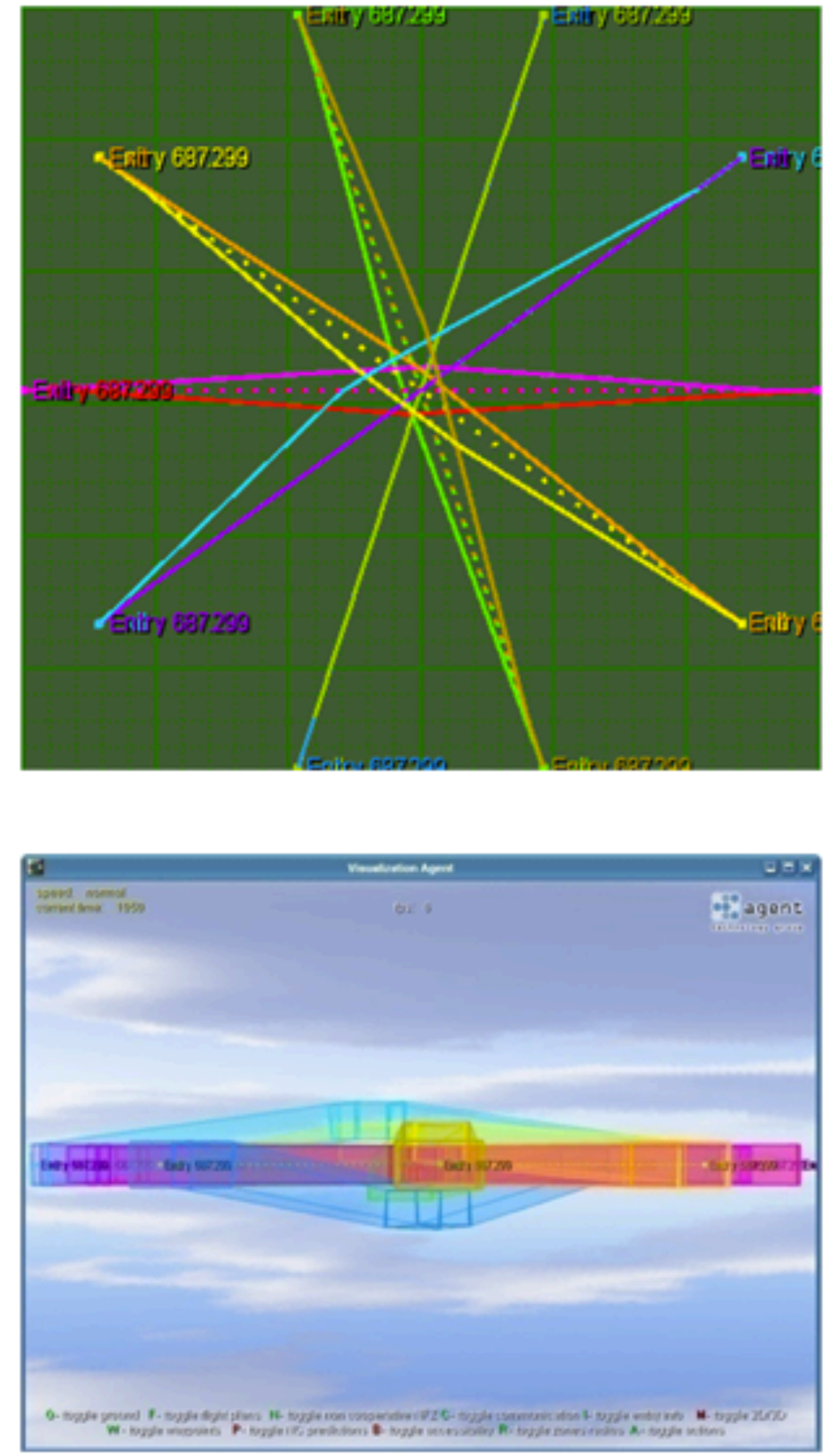
Deconfliction - comparison



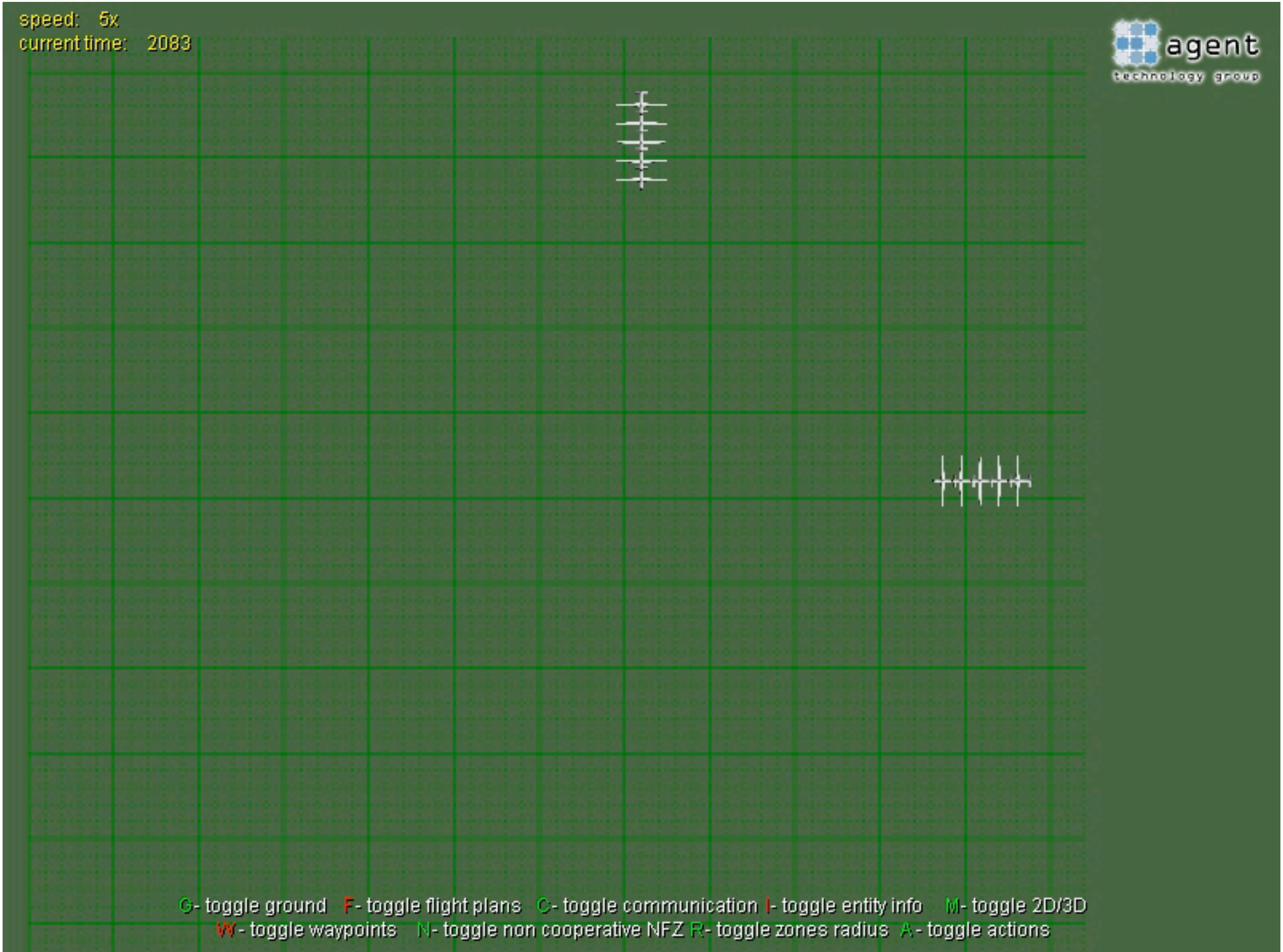
RBCA



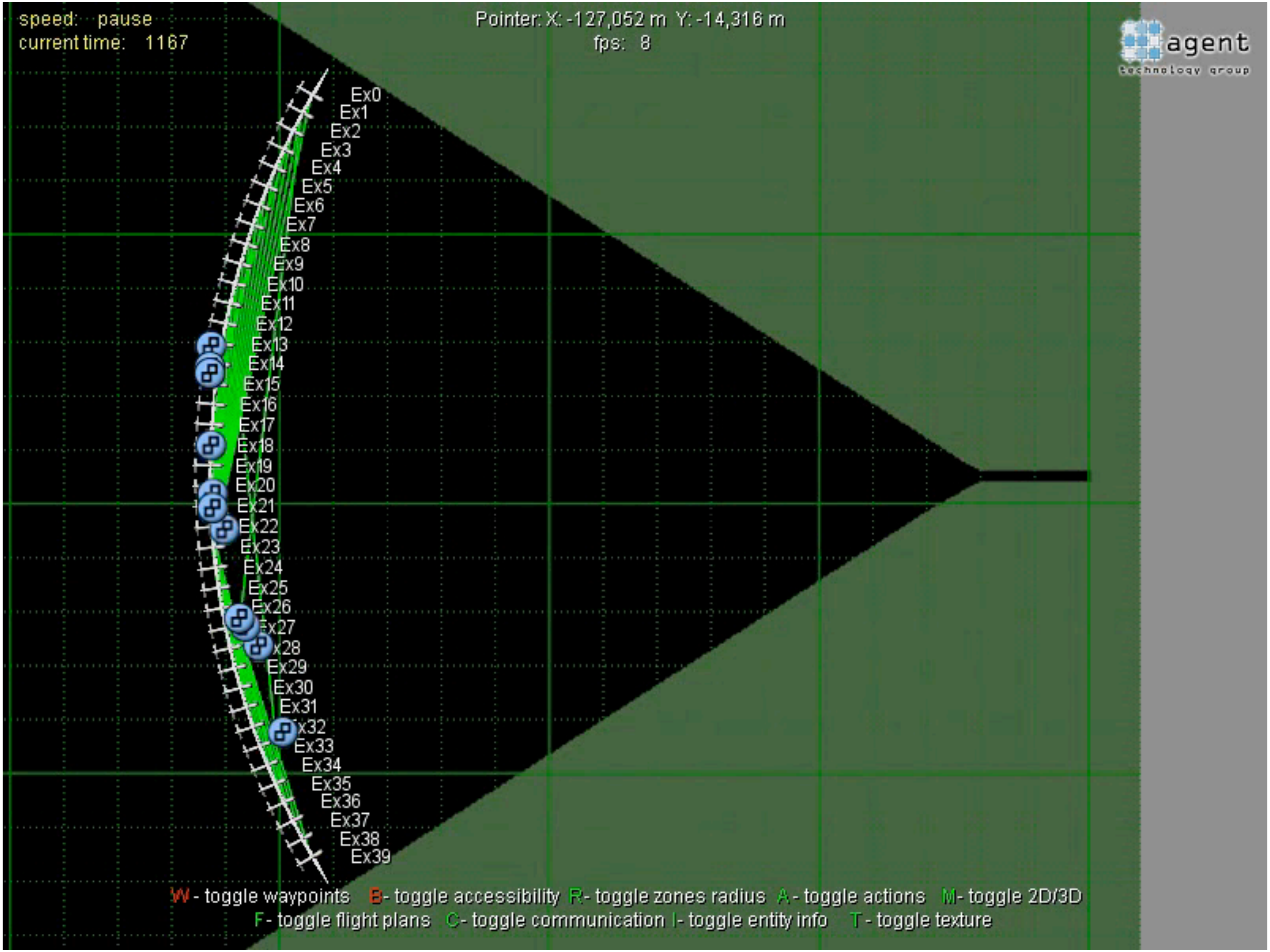
IPPCA



MPCA



speed: 5x
current time: 2083
agent
technology group
G- toggle ground F- toggle flight plans C- toggle communication I- toggle entity info M- toggle 2D/3D
W - toggle waypoints N- toggle non cooperative NFZ R- toggle zones radius A - toggle actions



speed: pause
current time: 1167
Pointer: X: -127,052 m Y: -14,316 m
fps: 8
agent
technology group
Ex0
Ex1
Ex2
Ex3
Ex4
Ex5
Ex6
Ex7
Ex8
Ex9
Ex10
Ex11
Ex12
Ex13
Ex14
Ex15
Ex16
Ex17
Ex18
Ex19
Ex20
Ex21
Ex22
Ex23
Ex24
Ex25
Ex26
Ex29
Ex30
Ex31
Ex33
Ex34
Ex35
Ex36
Ex37
Ex38
Ex39
W - toggle waypoints B - toggle accessibility R - toggle zones radius A - toggle actions M - toggle 2D/3D
F - toggle flight plans C - toggle communication I - toggle entity info T - toggle texture

Agent Concepts Apps Potential

- Coordination, negotiation, argumentation
- Game (cooperative/non-cooperative) theory, mechanism design, combinatorial auctions, adversarial reasoning
- Simulation and modeling
- Interoperability, Knowledge Sharing, Ontologies
- Adjustable Autonomy, Policies, Electronic institutions, Organization and Norms
- Distributed learning, distributed data analysis
- Distributed resource allocation, distributed Planning
- Trust and reputation
- Multiagent programming and formal verification of multiagent programs

Opportunities for Deployment

Distributed and collective aspects of agency

Autonomy oriented aspects of agency

Opportunities for Deployment

Distributed and collective **aspects of agency:**

- ***Decentralized scenarios:*** **where the data and knowledge required for computation are not or cannot be available centrally or the process physical system control needs to be distributed:**
  - *Geographical distribution* of knowledge and control (e.g., logistics, collaborative exploration, mobile and collective robotics, pervasive systems) or the environments with partial or temporary communication inaccessibility (where self-organization, local interaction and intelligent synchronization is needed in order to cope with communication inaccessibility).
  - *Competitive domains*, with the restrictions on the information sharing (e.g., e-commerce applications, supply-chain management, and e-business).
  - Domains with the requirements for *time-critical response* and *high robustness* in distributed environment (e.g., time–critical (soft- and/or hard-realtime) manufacturing or industrial systems control, with re-planning, or fast local reconfiguration).

Opportunities for Deployment

Distributed and collective aspects of agency:

- *Decentralized scenarios*: where the data and knowledge required for computation are not or cannot be available centrally or the process physical system control needs to be distributed
- *Simulation and modeling scenarios*: Using agents for simulation purposes has been very common, while the right justification was often missing. Agents shall be deployed in simulation exercises where we require, e.g., an easy migration from the simulation to deployment in real environment.
- *Open systems scenarios*: In scenarios requiring integration and interoperability among software systems that are not known a priori and whose source code may not be available – here the use of agent technologies, especially agent communication languages and interoperability standards is advisable.

Opportunities for Deployment

Distributed and collective aspects of agency

Autonomy oriented aspects of agency:

- appropriate in application domains with high requirements for systems with decision making autonomy, when the user delegates the substantial amount of decision making authority to the system and when the system is expected to cope independently with unexpected situations (also in the situation with long term communication inaccessibility and interaction isolation of the autonomous entity).

Agents Application Sectors

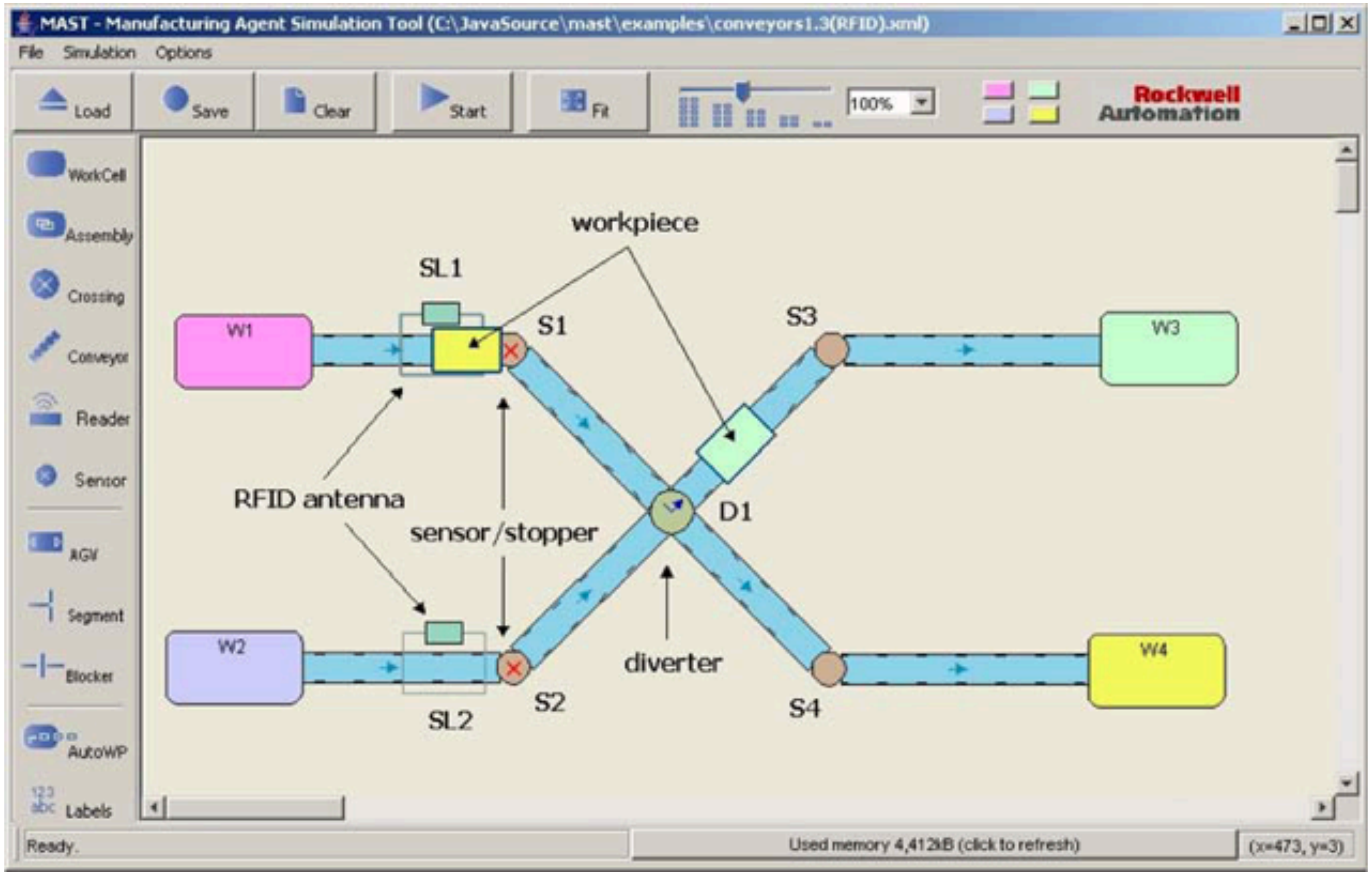
- Manufacturing and production
- Traffic and logistics
- Robotics, autonomous systems
- Air traffic and space
- Security applications
- Energy

Manufacturing and production planning

- Rockwell: RFID enabled material handling control
- Daimler: Production 2000+
- SkodaAuto: Production planning of engine assembling

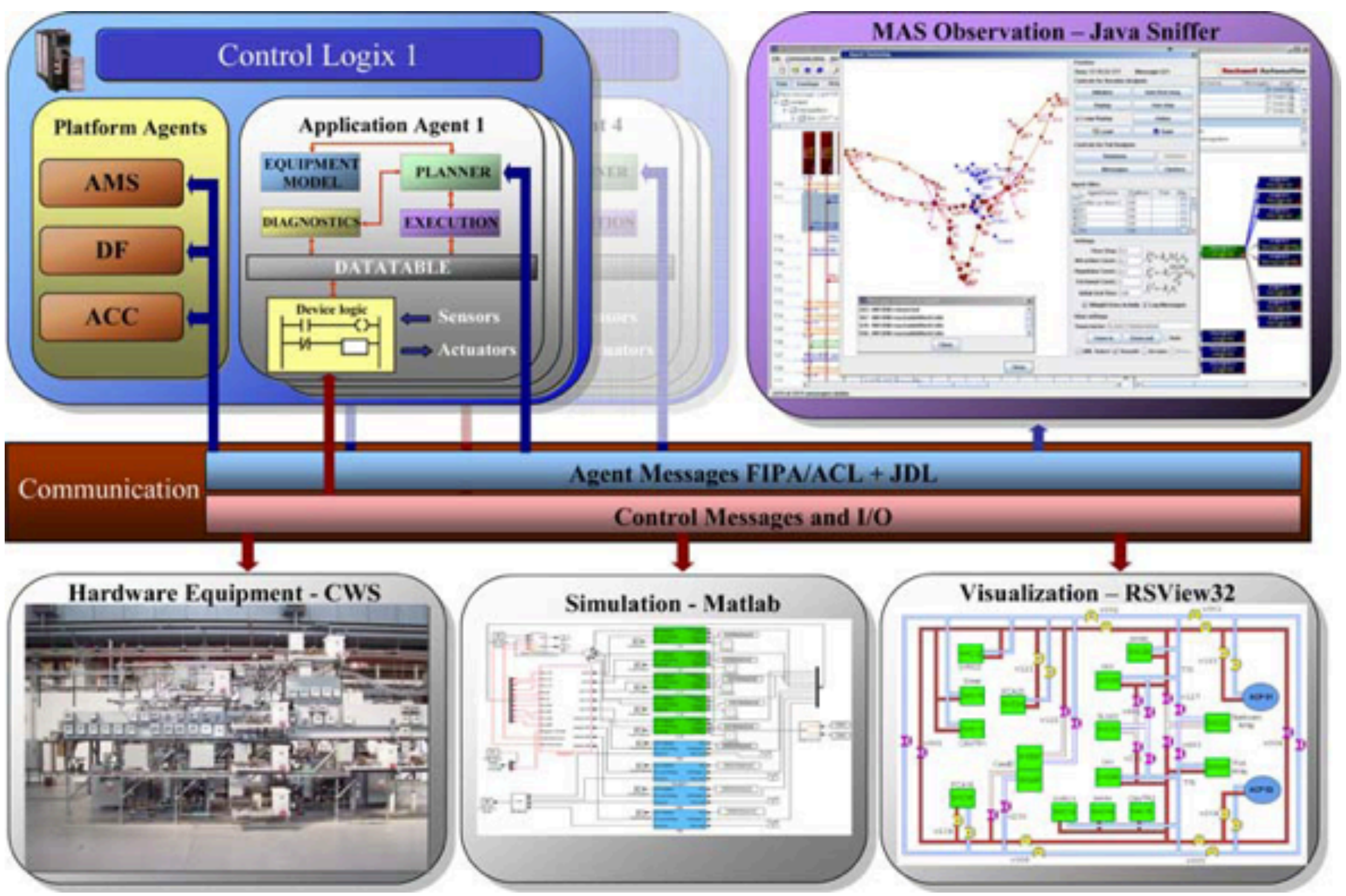
Manufacturing

- Rockwell: RFID enabled material handling control
  - Simulation tool for material handling in flexible manufacturing (in JADE)
  - Direct deployment towards to a hardware testbed provided by University of Cambridge and Vienna University of Technology, cooperation with Gilllette



Manufacturing

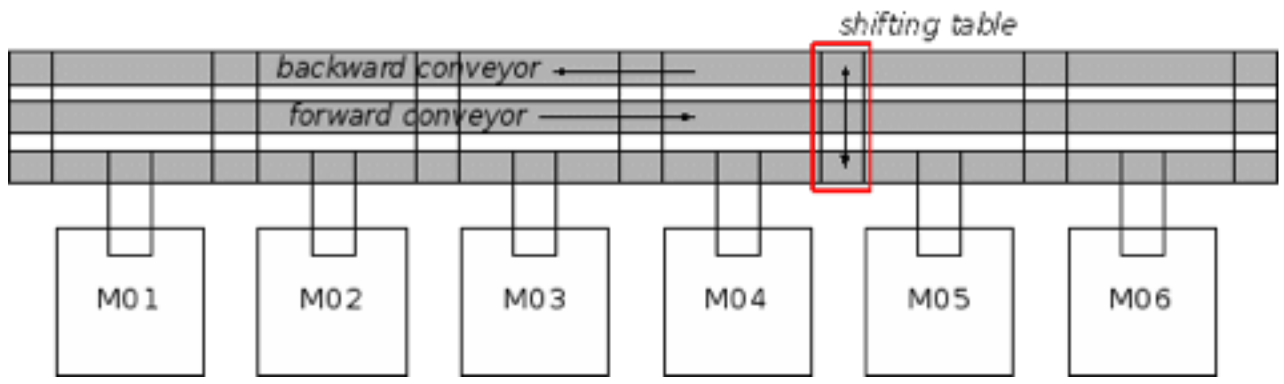
- Rockwell: <u>Autonomous Cooperative System (ACS)</u>
  - Use of the technology for shipboard automation distributed control and diagnostics
  - Deployment on a physical US Ship model

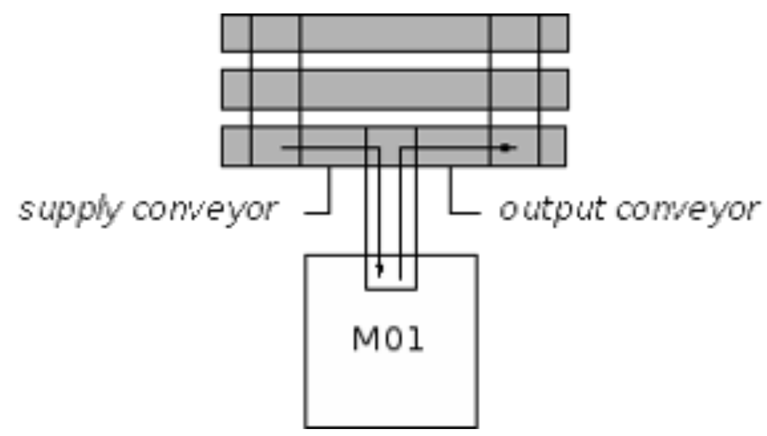


Manufacturing

DAIMLER

- Daimler: Production 2000+
  - System for control of flexible machines, flexible transportation systems
  - High volumes, flexibility and quality => minimize redundancy and overcapacity





  - KOWEST architecture: workpiece – machine – switch agents
  - Late commitment reactive approach, driven by the workpiece
  - Combinatorial auctioning for allocation

Manufacturing

DAIMLER

- Daimler: Production 2000+
  - Deployment towards manufacturing of OM 646 and OM 602 diesel engines
  - Robust operation for 5 years – 16 hours/day

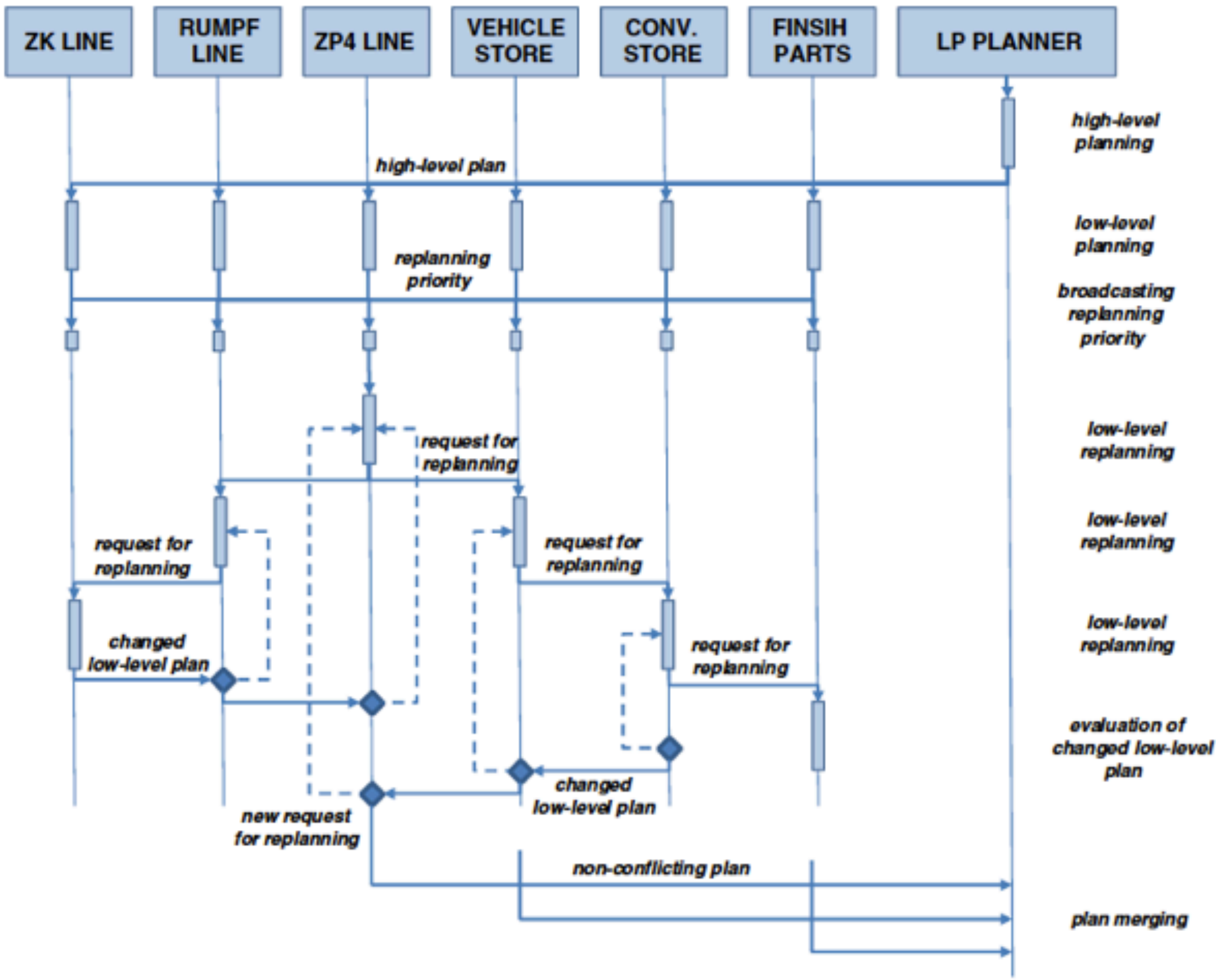
  - No further deployment beyond 2005. Investment into flexibility too high.
  - Flexibility requirements at Daimler have been limited

Manufacturing

- SkodaAuto: Production planning of engine assembling
  - The factory manufactures daily up to 2,000 pieces of engine heads for 2 and 4 cylinder 1,200cm$_3$ engines, 2,000 pieces of 3 cylinder RUMPF engines, or 1,200 pieces of finished 3 and 4 cylinder 1,200cm$_3$ engines.
  - Delivered in cooperation of Czech Technical University and T-Systems
  - Combination of high-level (linear-programming-based) planning and low-level (negotiation-based) planning.

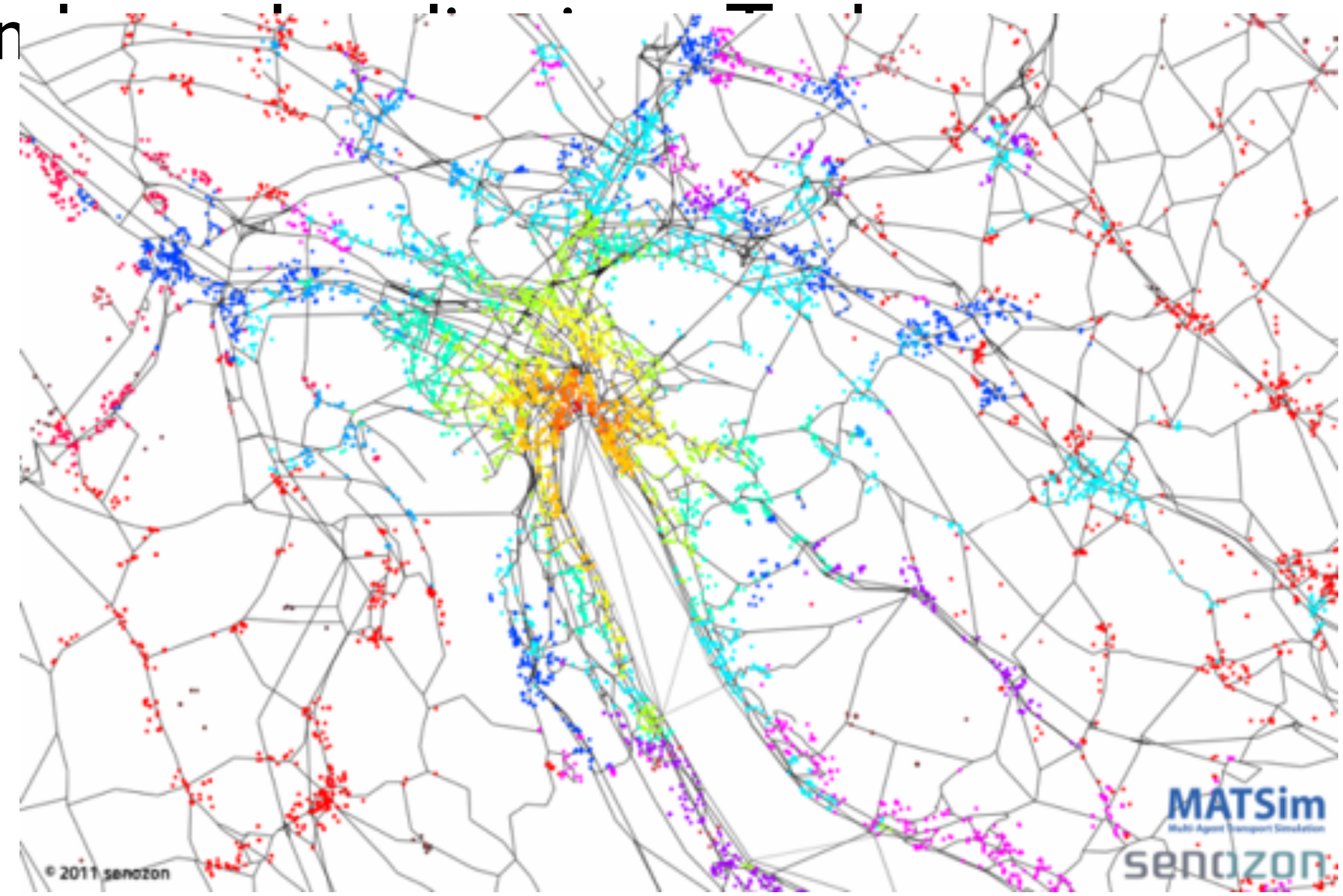
Manufacturing

- SkodaAuto: Production planning of engine assembling



Traffic and Logistic

- (–) limited deployment on routing vehicles
- (+) often used as an suboptimal optimization approach
  - reactive negotiation, anytime replanning
  - large scale simulation
- Whitestein/LivingSystems – logistics planning for DHL Freight dispatching, deployed in 17 European countries, 100 subsidiaries and 1200 users.
- MAGENTA - Private car hire an
  international, GIST Logistics,
- MATSIM/SENOZON – multiagent transport simulation:
  - Zurich, Switzerland
  - Padang, Indonesia
  - Gauteng, South Africa
  - Toronto, Canada
  - Berlin, Germany

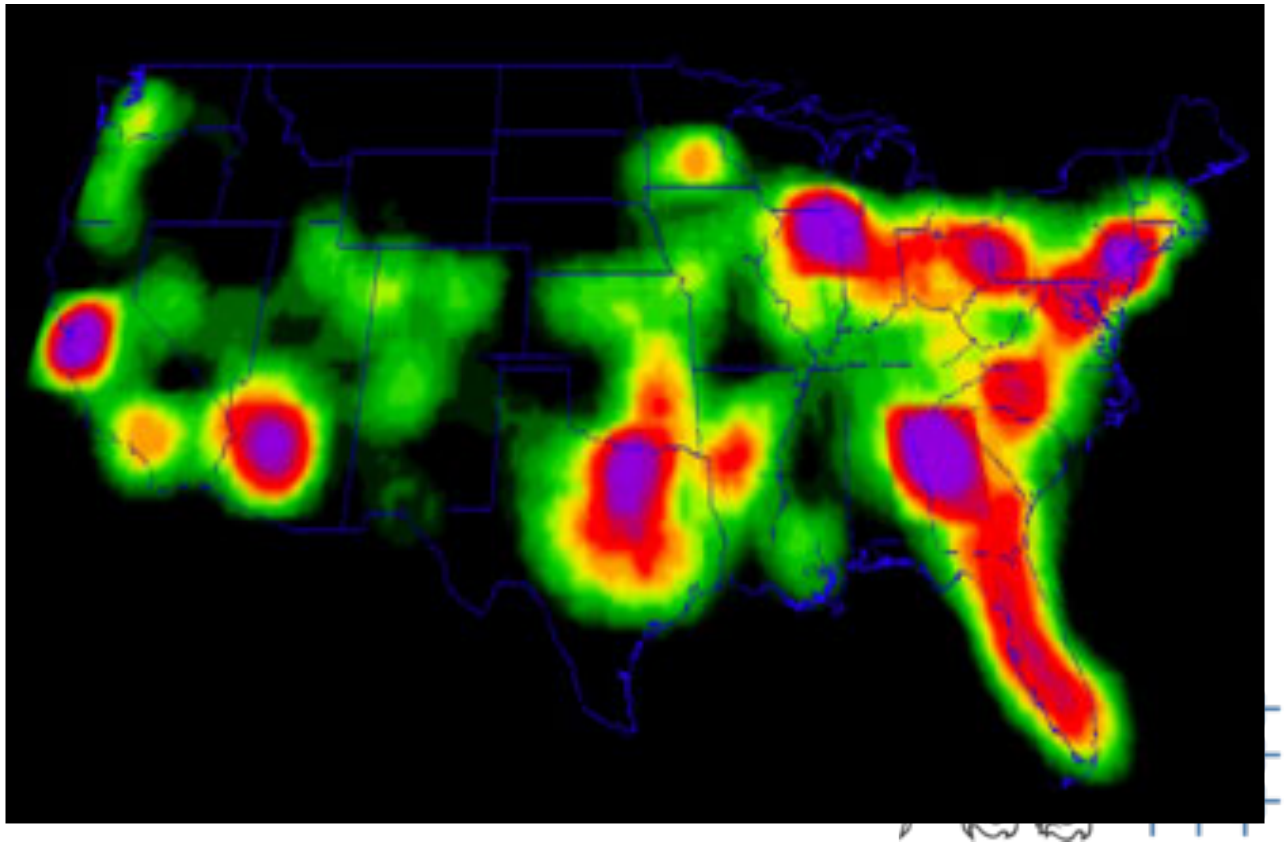


Air Traffic

- NASA: ACES (Advanced Concepts Evaluation System)
- LOSTWAX: Areogility
- ATG: AgentFly
- FL3XX

Air Traffic

- NASA: ACES (Advanced Concepts Evaluation System)
  - multi-fidelity non-real-time modeling and simulation system with full gate-to-gate representation of all the major components of the NAS
  - the agent-based modeling approach that is being used represents the individual behaviors of the airspace participants and captures the critical ripple effect of one user's actions on other system participants
  - based on CybelePro robust high-performance infrastructure, services and tools for rapid development and deployment of large-scale distributed systems and integrated solutions

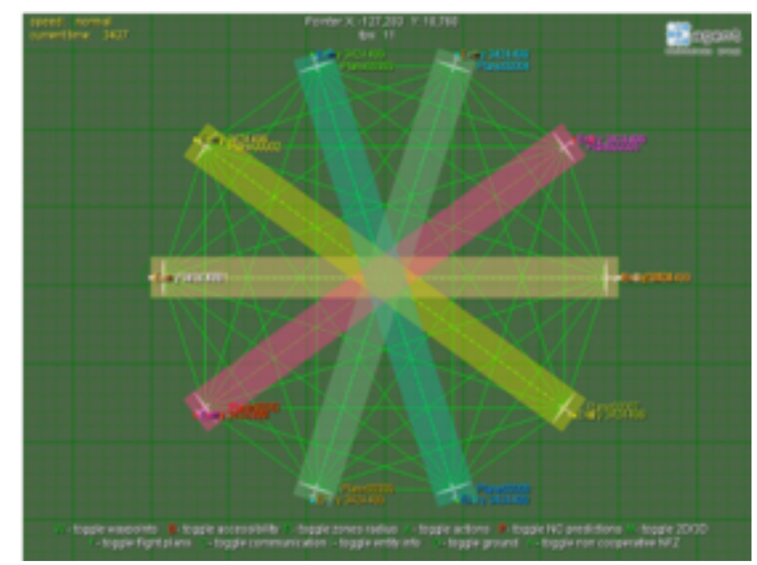


Air Traffic

- LOSTWAX: Areogility
  - multiagent decision support system for modeling of the jet engine aftermarket (provided to Rolls Royce)
  - used for supply-chain and inventory optimization
  - used for costing of leasing services of the jet-engines
  - extended to business process modeling beyond aerospace (defense)

Air Traffic

- ATG: <u>AgentFly</u> – Free-flight planning and re-planning of collision free trajectory for multiple aerial autonomous systems:
  - decentralized planning, collision avoidance, multiagent simulation
  - extended for NAS modeling
  - based on AGLOBE multiagent OS



Air Traffic

- FL3XX: An integrated approach to flight operations that reduces your costs
  - aimed at small private jets,
  - planning as a service
  - based on AGLOBE multiagent OS

Security applications

Deployment of game-theoretical and adversarial reasoning methods in

- Infrastructure security
- Cybersecurity

Security applications

- **Infrastructure security:**
  - leading research center – TEAMCORE, University of Southern California
  - Game-theoretical randomization of defense strategies:
    - ∗*ARMOR -- LAX international airport inspection checkpoints*
    - ∗*IRIS – randomizes schedules for Federal Air Marshals*
    - ∗*PROTECT – route planning for Coast Guards in port of Boston*
    - ∗*RAIL – scheduling of fare-evasion inspections in LA metro*
    - ∗*Urban security in the city of Mumbai*
  - also Agent Technology Center (CTU):
    - ∗*AGENTC – Maritime security in piracy environment in Gulf of Aden*


Security applications

- **Cybersecurity:**
  - CognitiveSecurity (Agent Technology Center startup):
  - Network Behavior Analysis, based on statistical analysis of NETFLOW data
  - Challenges:
    - *Lower false positive rate, provide self-adaptation and prevent strategic overtraining*
  - Agents methods deployed:
    - *Trust modeling, agent-based simulation of attacks, game-theoretic modeling of the intruder*
  - Product on the market

cognitivesecurity

Energy

- Modeling and optimization
- Decentralized resource allocation

Energy

- **Modeling and optimization:**
  - NODA Intelligent Systems – startup from PhD thesis: Multi-Agent Systems for District Heating Management, Fredrik Wernstedt
    - *Design of decentralized control and monitoring architecture*
    - *EON Heat, Landskronahem, Karlshamn Energi, BA Construction*

- **Decentralized resource allocation:**
  - IDEaS – SOTON project exploring distributed control, operation and management in future electricity networks
  - Connects producers and consumers of energy via dynamically negotiated contracts
  - Industrially funded

Other?

- Film industry – Massive software

Analysis

Domain	Concept	Maturity	integration	Functionality	Platform	Company
Manufacturing	Simulation, coordination, resource allocation interoperability	Demonstrator, prototype	HW Critical, legacy	Planning, control, simulation	Proprietary, JADE	Rockwell, Daimler Skoda, T-Systems
Logistics, transport	Resource allocation, simulation	Applications	No, limited	Planning, simulation	Proprietary	Whitestein, Magenta, Senozon
Air Traffic	Simulation, coordination	Application	HW (UAV)	Planning, Simulation	CybelePRO, AGLOBE	NASA, AgentFly, Aerogility, FI3xx
Security	Game theory	Demonstrator, Application	No	Resource allocation, scheduling	No	governments
Energy	Simulation, coordination, Game theory	Application, concepts	HW	Planning, resource allocation	No	NODA, ..

Deployment Bottlenecks

Intelligent systems based application still do not fully leverage the potential of the multi-agent technologies due to:

- **Potential still not believable** – lack of case-studies, lack of convincing quantitative argument
- **Limits on scalability and fidelity** – early demos are not believed to scale up to industrial requirements
- **High costs & high risks** – deployment is costy and benefits not certain

ATG R&D Activities

Air Traffic Management	Unmanned Aerial Syst	Critical Infrastructures	Cybersecurity
Mid term deconfliction	Sense & avoid	Game theory &	Ensamble classification
Trajectory driven ATM	Tactical planning	Adversarial planning	Network behaviour analysis
NAS modeling	Trajectory planning	Maritime security	Opponent modeling
ATC modeling	HW deployment	Fare evasion in LA	Adversary evasion
	Mixed reality sim	Traffic & mobility nets	Steganography
	Human/Machine interf.		Social networks privacy