Quick Quiz …

"**Developing Intelligent Agent Systems: A Practical Guide**" was published in …

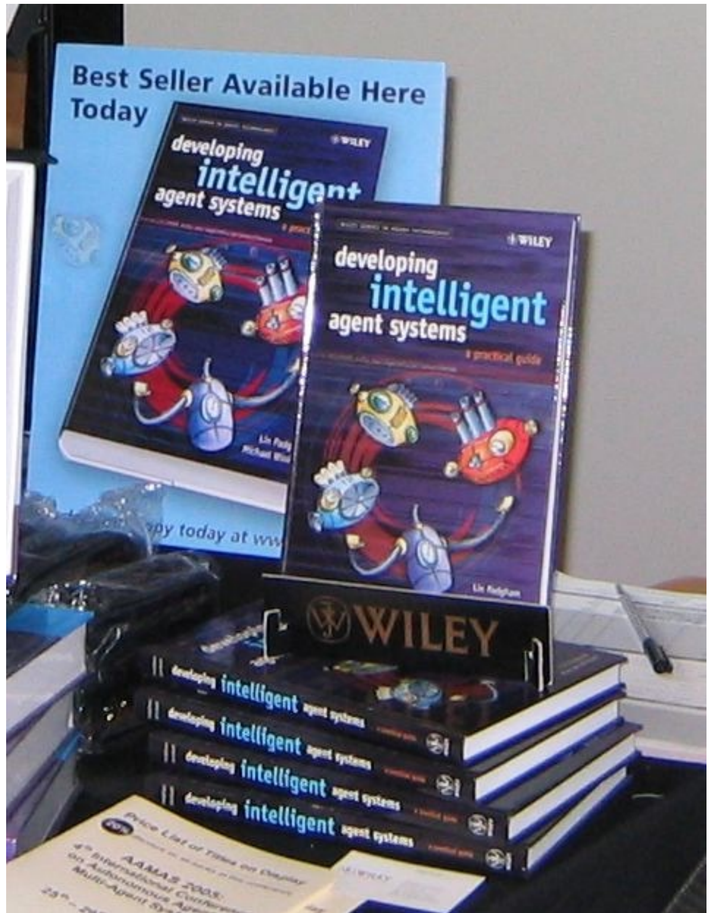
- 2024
- 2021
- 2018
- 2014
- 2004
- 1993

Quick Quiz …

"**Developing Intelligent Agent Systems: A Practical Guide**" was published in …

- 2024
- 2021
- 2018
- 2014
- **2004**
- 1993

Lin Padgham and Michael Winikoff. Developing Intelligent Agent Systems: A Practical Guide. June 2004, ISBN 0-470-86120-7, John Wiley and Sons.



Quick Quiz …

"**Agent-Oriented Programming**" was published in …

- 2024
- 2021
- 2018
- 2014
- 2004
- 1993

Quick Quiz …

"**Agent-Oriented Programming**" was published in …

- 2024
- 2021
- 2018
- 2014
- 2004
- **1993**

Yoav Shoham, Agent-oriented programming, Artificial Intelligence, Volume 60, Issue 1, 1993, Pages 51-92, https://doi.org/10.1016/0004-3702(93)90034-9.

Abstract: A new computational framework is presented, called agent-oriented programming (AOP), which can be viewed as a specialization of object-oriented programming. The state of an agent consists of components such as beliefs, decisions, capabilities, and obligations; for this reason the state of an agent is called its mental state. …

30 Years of Engineering Multi-Agent Systems: Lessons for Agentic systems

Michael Winikoff - 27 Nov 2025 - winikoff.github.io

Link to these slides:
winikoff.github.io/pdf/DXC.pdf

(This link will also be on the last few slides)

Aim: Introduction & Orientation

Outline:

- LLMs and agentic software
- **Introduction**: IFAAMAS, AAMAS, EMAS, “agent”
- **Orientation**: Engineering Multi-Agent Systems (EMAS): what, why, state-of-the-art
- EMAS & LLMs
- (Briefly) non-engineering agent work and LLM relevance

Link to these slides:
winikoff.github.io/pdf/DXC.pdf

Large Language Models (LLMs)

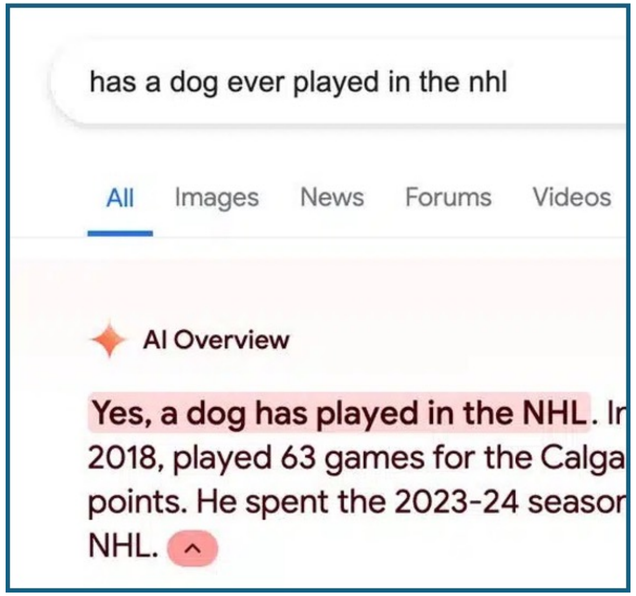
Q: How many fingers?
A: five
(e.g. from Dr Haibo Yang)

- Opaque
- "Bullshit" (aka "hallucination")
- Leakage
- Lack of reasoning (next slide)
- Weak planning

(Valmeekam et al 2024)

- Energy
- IP theft

Original audio: And oops, by accident, the ball goes through the window of his house.
Transcribed: And oops, by accident, the ball goes through the window of his house. So when Christina walks over and says, Miss, I want you to give a dollar to me, I mean, it has essence nothing more!



Sources: https://searchengineland.com/google-ai-overview-fails-442575
https://doi.org/10.1145/3630106.3658996

Valmeekam, K., Stechly, K., Kambhampati, S.: LLMs Still Can't Plan; Can LRMs? A Preliminary Evaluation of OpenAI's o1 on PlanBench (Sep 2024). https://doi.org/qdkr

AI Overview +2

If at 10 AM you set your clock to noon, it will show **12:00 PM** at noon and will show 1:00 PM at 1 PM, because you've just set it forward two hours, making it accurately reflect the current time.

Here's the breakdown:

1. **At 10:00 AM:** You manually set your clock forward by two hours to 12:00 PM (noon).
2. **At 12:00 PM (Noon):** The clock shows 12:00 PM because that is the correct time when it is noon.
3. **At 1:00 PM:** After another hour passes, the clock will have advanced normally, showing 1:00 PM.

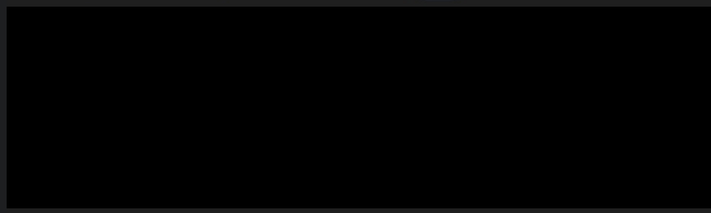
AI Overview +6

No, it is not possible for one identical twin to be aware of the other's existence while the other is not. Identical twins are born at the same time from a single fertilized egg and share the same placenta and amniotic sac, so they are always aware of each other's existence from birth.

- If two identical twins are separated at birth, both would be unaware of the other's existence unless they are later reunited.
- There is no scientific evidence to support the idea of "twin telepathy," where twins can sense each other's emotions or thoughts without any other form of communication.

AI Overview +3

If you watch episode 2 after episode 1 and before episode 3, **you might miss some character development or plot points that were introduced in episode 1** but the experience is likely to still be understandable for most shows. For a standard TV show, watching episodes in order is generally recommended, but a slightly out-of-order viewing will probably not ruin the experience, especially if the story is easy to follow.



AI Overview +1

If you take two bags of trash to the curb every week for a year, you will have a total of **104 bags** of trash at the curb after one year.

Source: https://bsky.app/profile/conitzer.bsky.social

Agentic Systems

- Large Language Models (e.g. ChatGPT, Claude) as autonomous agents
- Highly hyped!
- But given LLMs' (aka "stochastic parrot") lack of understanding and tendency to hallucinate and "over-optimise" (below), not a great idea!
- Ongoing work on providing infrastructure to restrain LLMs

```
121
122 # Verify packages were created and check sizes
123 echo "Verifying Lambda packages were created..."
124 echo "✓ All Lambda packages verified successfully"
125 # Step 3: Deploy using Multi-Stack Approach
```


Source: https://mastodon.nz/@paco@infosec.exchange/115151810517438958

Using LLMs to create agent(ic) software

- A ***partnership*** between software and human
  - LLM needs to be able to explain: what it did, why, what it can(not) do effectively
  - Human needs to: decompose problem, specify requirements (how?) & tests, prompt the LLM, review generated code … and write (some) code (Ullrich *et al.*, 2025)
- Key challenges include **guidance** (methodologies), and **reliability/safety**
  - Role for hybrid architectures
  - Enter EMAS …

Ullrich, J., Koch, M., Vogelsang, A.: From requirements to code: Understanding developer practices in LLM-assisted software engineering (2025), https://arxiv.org/abs/2507.07548

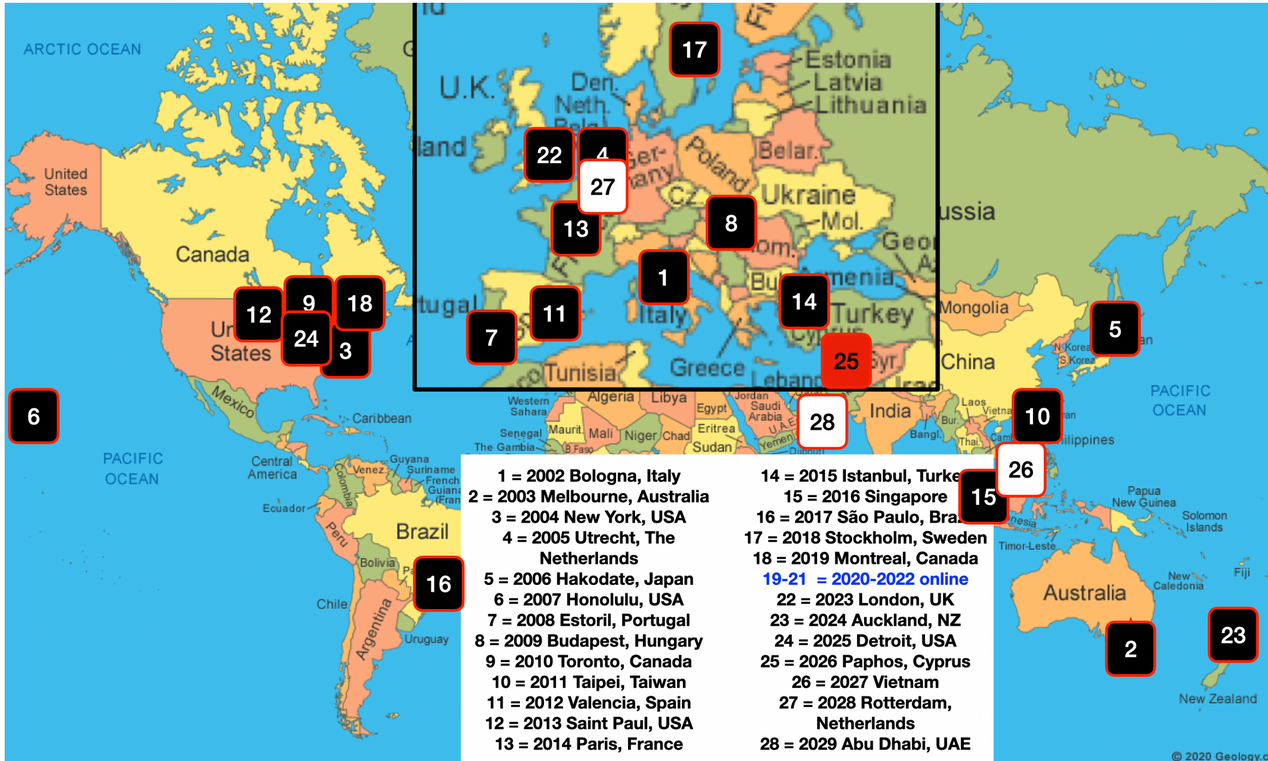
IFAAMAS? AAMAS? EMAS?

- IFAAMAS - umbrella organisation (next slide) [IF = International Foundation]
- AAMAS - main conference - diverse topics, including EMAS
- JAAMAS - main journal (not under IFAAMAS)
- **Today focus mostly on EMAS**

MAS = Multi-Agent Systems

EMAS = Engineering MAS

AAMAS = Autonomous Agent and MAS



IFAAMAS

Who

- IFAAMAS is a non-profit corporation, chartered in the US.
- It is a membership organization: Anyone who has attended 2 out of the 4 most recent AAMAS conferences (ignoring 2020) is a member
- Board of Directors (volunteers!):
  - 27 people, elected by the membership (list on next slide)
  - Various committees (e.g. awards, conference bids)
  - Each serves a 6 year term, and cannot be immediately re- elected
  - Elections every 2 years thus replace 1/3 of the Board

What

- Ensure the AAMAS conference continues
  - Selects venues, selects chairs, provides seed funds, underwrites contracts, establishes policies, …
- More broadly, promotes education, research, application and awareness of autonomous agents and multiagent systems
  - Supports student participation at AAMAS
  - Supports “Agent Schools”
  - Creates rewards to recognize and encourage quality research
  - …

“In contrast to remote control, this sophisticated set of computer programs acts as an agent of the operations team on board the remote spacecraft. Rather than have humans do the detailed planning necessary to carry out desired tasks, remote agent will formulate its own plans, using high level goals provided by the operations team. Remote agent devises its plan by combining those goals with its detailed knowledge of both the condition of the spacecraft and how to control it.” http://nmp.nasa.gov/ds1/tech/autora.html

https://web.archive.org/web/20150910115102/https://nmp.nasa.gov/ds1/tech/autora.html

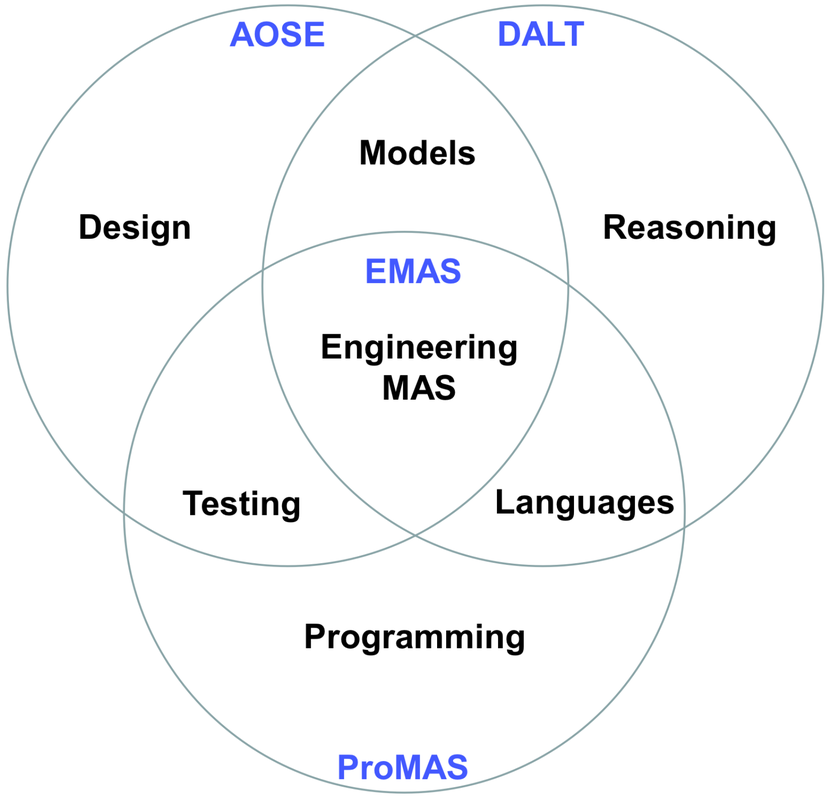
Deep Space 1 is an *agent*

- **Situated** (“placed on board”)
- **Autonomous** (“In contrast to remote control”)
- **Proactive** (“… formulate its own plans, using high level goals”)
- **Reactive** (“If problems develop, remote agent in many cases will be able to fix them or work around them. …”)
- **Social** (“… If it cannot, it can request help from its sentient terrestrial collaborators”)

Also Multi-agent systems (MAS) and agent societies.

What is EMAS?

- Research on Engineering Multi-Agent Systems is concerned with a range of topics that aim to provide software engineers with the **concepts**, **processes**, **notations**, **techniques**, **languages**, and **tools** to be able to effectively develop multi-agent systems.
- *"Despite the substantial body of knowledge and expertise developed in the design and development of Multi-Agent Systems (MAS), the **systematic development of large-scale and open MAS** still poses many challenges"* (EMAS 2024 CFP, emphasis added)



EMAS 2014 keynote:
Twenty Years of Engineering MAS (Hindriks, 2014)

Koen V. Hindriks. 2014. The Shaping of the Agent-Oriented Mindset. EMAS. https://doi.org/mtbb Slides: https://emas14.wordpress.com/wp-content/uploads/2013/11/koen-emas-14.pdf

https://emas.in.tu-clausthal.de/2024/cfp/

Why?

Why (multi-)agents?

- Wide range of applications involving distribution, adaptability, flexibility, robustness, autonomy, complexity
- Domains include production scheduling, energy, transport, disaster relief, manufacturing, simulation, health, UAVs ...

Benfield et al, 2006
Belecheanu et al, 2006
Munroe et al, 2006
Dignum & Dignum, 2010
Müller & Fischer, 2014
Singh & Padgham, 2015
Briola et al, 2023
(Citation details next slide)

Why Engineering?

- Why not just write it Java?
- What about existing methodologies? ...

Source: Singh & Padgham, 2015

Citations

- Steve S. Benfield, Jim Hendrickson, and Daniel Galanti. 2006. Making a strong business case for multiagent technology. AAMAS. https://doi.org/cspvtz
- Roxana A. Belecheanu, Steve Munroe, Michael Luck, Terry Payne, Tim Miller, Peter McBurney, and Michal Pěchouček. 2006. Commercial applications of agents: lessons, experiences and challenges. AAMAS. https://doi.org/c3t6x5
- Steve Munroe, Tim Miller, Roxana Belecheanu, Michal Pěchouček, Peter McBurney, and Michael Luck. 2017. Crossing the agent technology chasm: Lessons, experiences and challenges in commercial applications of agents. The Knowledge Engineering Review. http://doi.org/b6xbv3
- Virginia Dignum and Frank Dignum. 2010. Designing agent systems: state of the practice. https://doi.org/fnqs7n
- Jörg P. Müller and Klaus Fischer. 2014. Application Impact of Multi-agent Systems and Technologies: A Survey. In: Shehory, O., Sturm, A. (eds) Agent-Oriented Software Engineering. Springer, Berlin, Heidelberg. https://doi.org/mtbg
- Dhirendra Singh and Lin Padgham. 2015. Community Evacuation Planning for Bushfires Using Agent-Based Simulation: Demonstration. AAMAS. https://www.ifaamas.org/Proceedings/aamas2015/aamas/p1903.pdf
- Daniela Briola, Angelo Ferrando, and Viviana Mascardi. 2023. Fantastic MASs and Where to Find Them: First Results and Lesson Learned. EMAS. https://doi.org/ms97

Benefits?

"Based on this analysis, agent-oriented methodology, architecture, and development delivered a 368% improvement on overall project productivity. (2.11 FP Actual/0.45 FP Expected) …

In a **wide range of complex business applications**, we show that the use of **BDI** [belief-desire-intention] **technology** incorporated within an enterprise-level architecture can **improve** overall developer **productivity** by an average [of] 350%. For java coding alone, the increase in productivity was over" (Benfield et al., 2006; emphasis added)

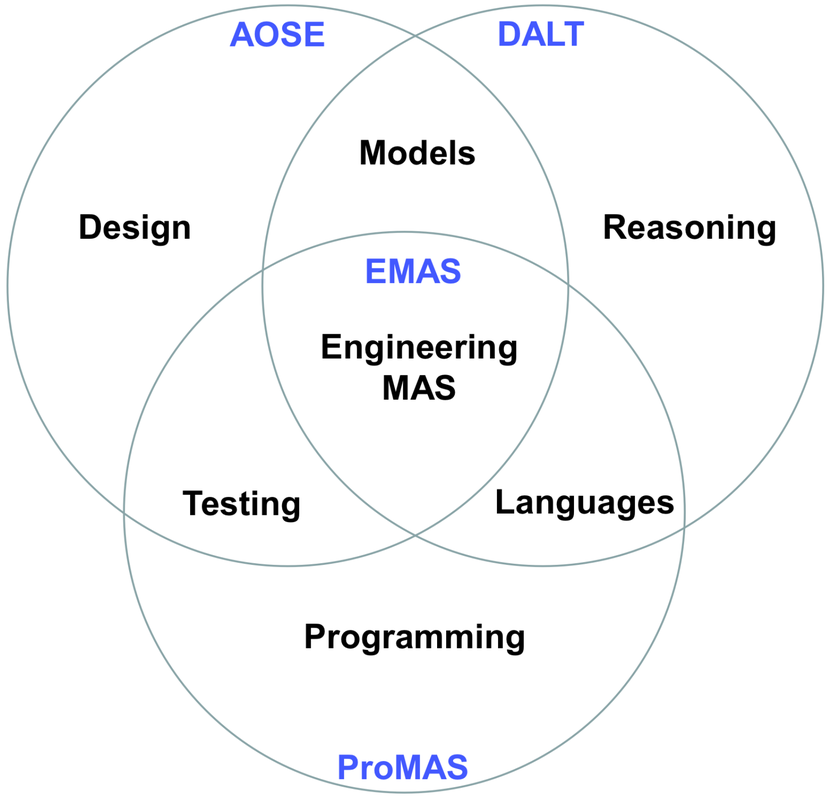
- Paper by Agentis Software staff — company commercialising agent technology, industries include finance, insurance, logistics, energy, IT infrastructure.

But: unverified claims by industry …

Steve S. Benfield, Jim Hendrickson, and Daniel Galanti. 2006. Making a strong business case for multiagent technology. AAMAS. https://doi.org/cspvtz

What is EMAS?

- Research on Engineering Multi-Agent Systems is concerned with a range of topics that aim to provide software engineers with the **concepts**, **processes**, **notations**, **techniques**, **languages**, and **tools** to be able to effectively develop multi-agent systems.
- *"Despite the substantial body of knowledge and expertise developed in the design and development of Multi-Agent Systems (MAS), the **systematic development of large-scale and open MAS** still poses many challenges"* (EMAS 2024 CFP, emphasis added)



EMAS 2014 keynote:
Twenty Years of Engineering MAS
(Hindriks, 2014)

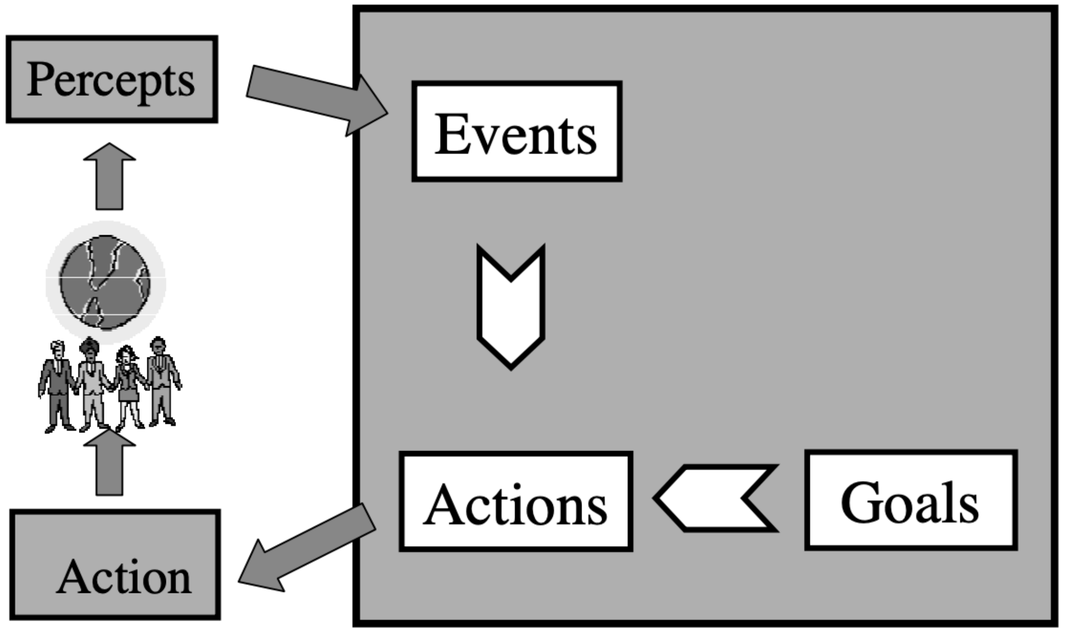
Koen V. Hindriks. 2014. The Shaping of the Agent-Oriented Mindset. EMAS. https://doi.org/mtbb Slides: https://emas14.wordpress.com/wp-content/uploads/2013/11/koen-emas-14.pdf

https://emas.in.tu-clausthal.de/2024/cfp/

Concepts

Agents are: (*property* - **concept**)

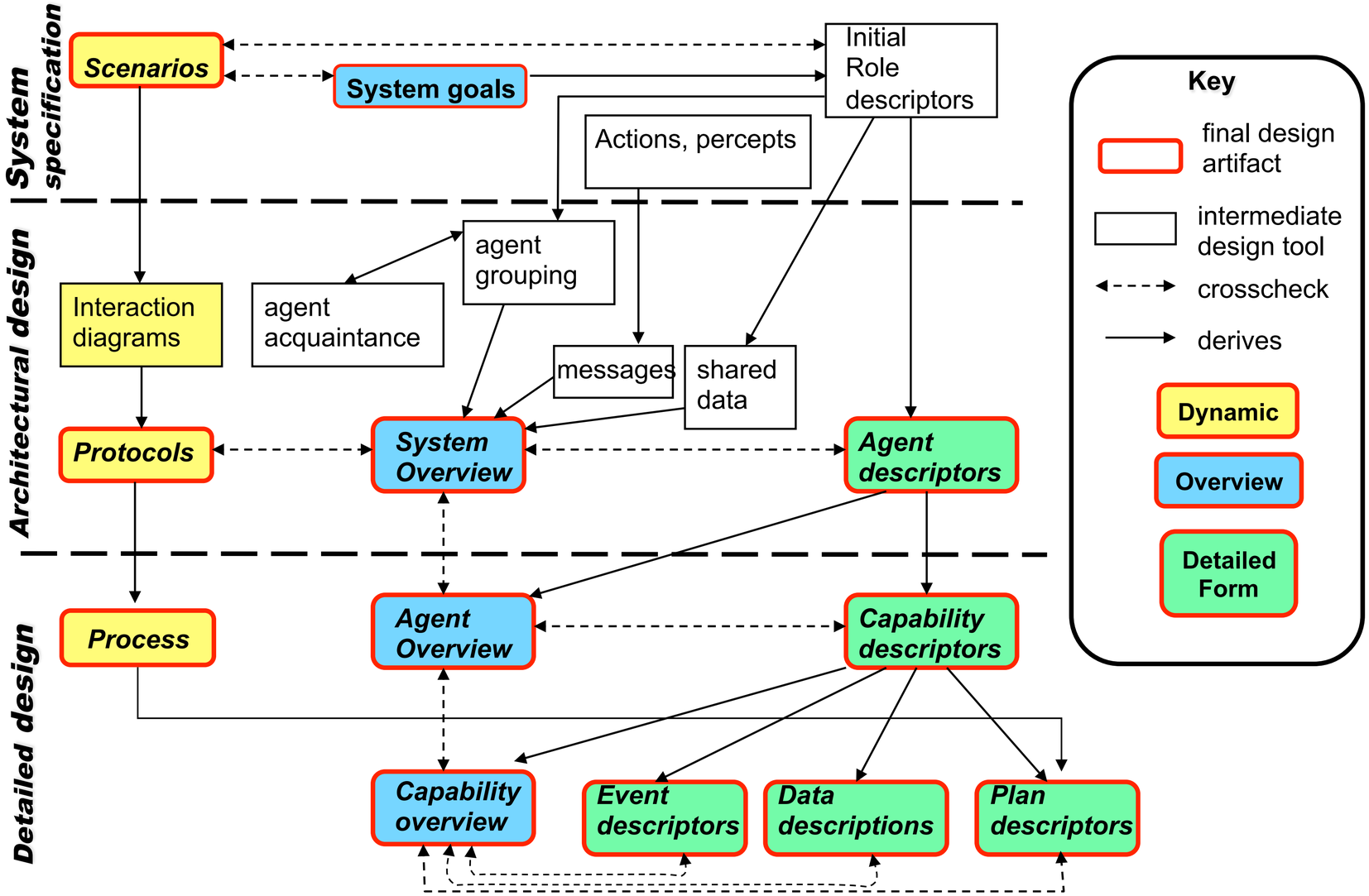
- *situated* - hence **actions** and **percepts**
- *reactive* - hence **events**
- *proactive* - hence **goals** (key concept)!
- *social* - hence **messages**, **protocols**, **commitments**, ...



Source: Winikoff et al, 2001

Michael Winikoff, Lin Padgham, and James Harland. 2001. Simplifying the Development of Intelligent Agents. AI. https://doi.org/dvbqgb

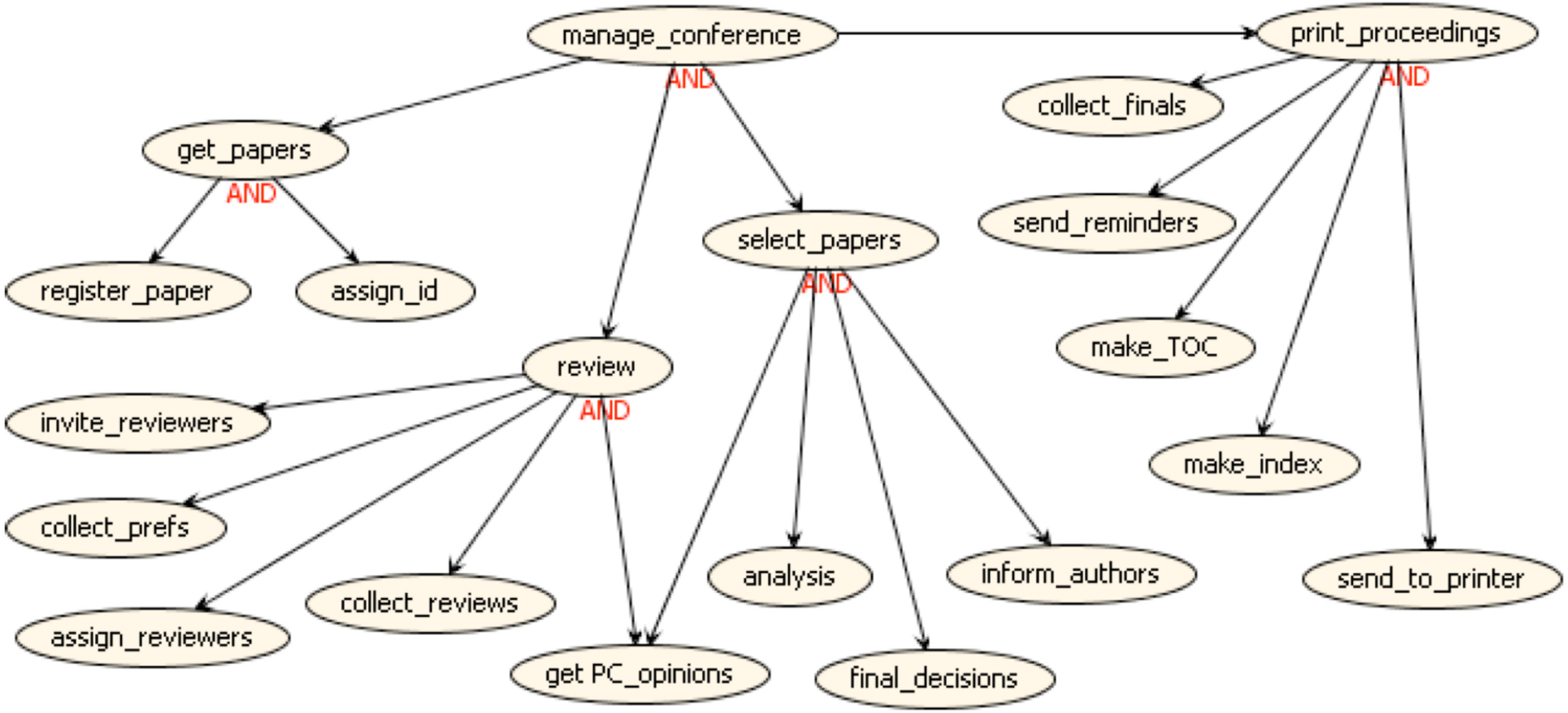
Process



- Also ...
- Implementation
- Assurance (testing, debugging, verification)
- Maintenance
- *Note: not sequential - iterative ... (and Agile ...)*

Lin Padgham and Michael Winikoff. Developing Intelligent Agent Systems: A Practical Guide. June 2004, ISBN 0-470-86120-7, John Wiley and Sons.

Example Goal Model/Notation



Source: Padgham et al, 2008

Lin Padgham, John Thangarajah, and Michael Winikoff. 2008. The Prometheus Design Tool – A Conference Management System Case Study. AOSE. https://doi.org/fn7r5q

Example User and System Stories

"*As* drone operator, *I want to* to be able to assign to the drones areas to explore, *so that* they can locate victims and notify me." (User story)

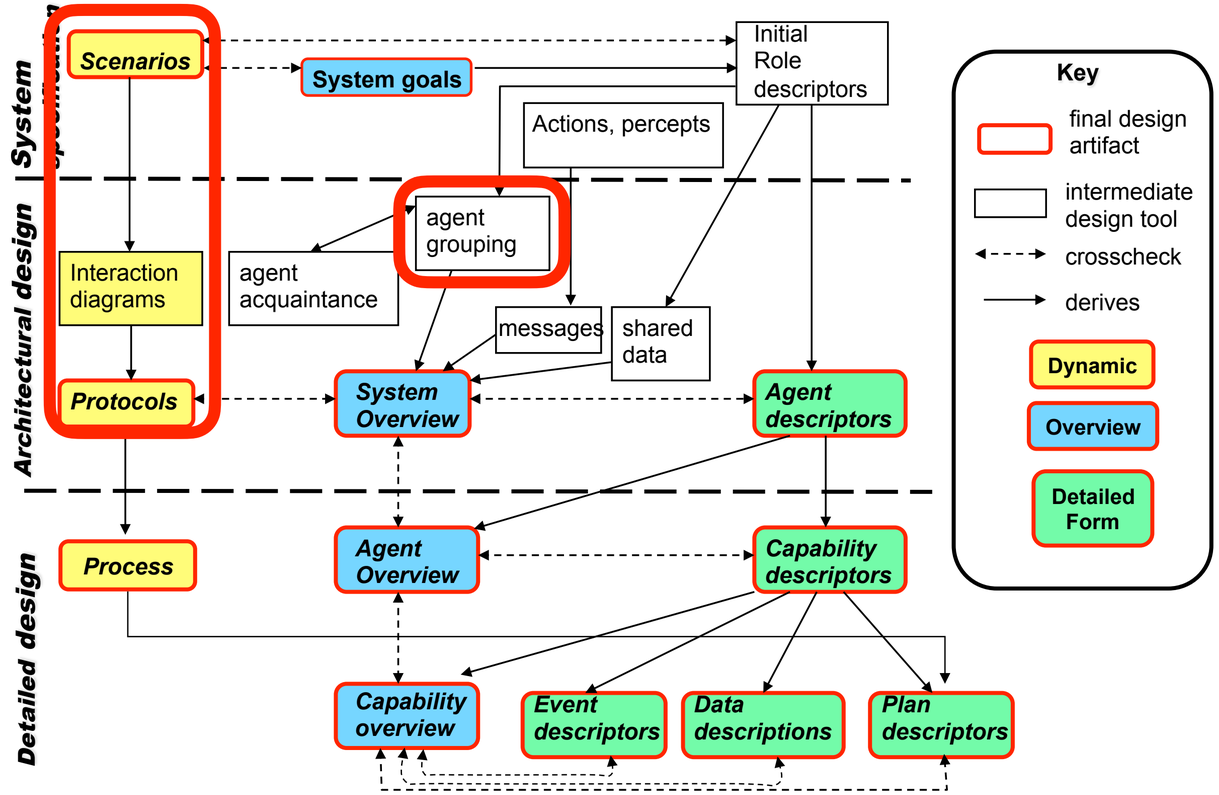
"*As* Drone, *I want to* explore an area assigned to me, *so that* I can identify victims." (System story)

Source: Rodriguez et al, 2021

Sebastian Rodriguez, John Thangarajah and Michael Winikoff. User and System Stories: an agile approach for managing requirements in AOSE. AAMAS-2021, pages 1064-1072. https://www.ifaamas.org/Proceedings/aamas2021/pdfs/p1064.pdf

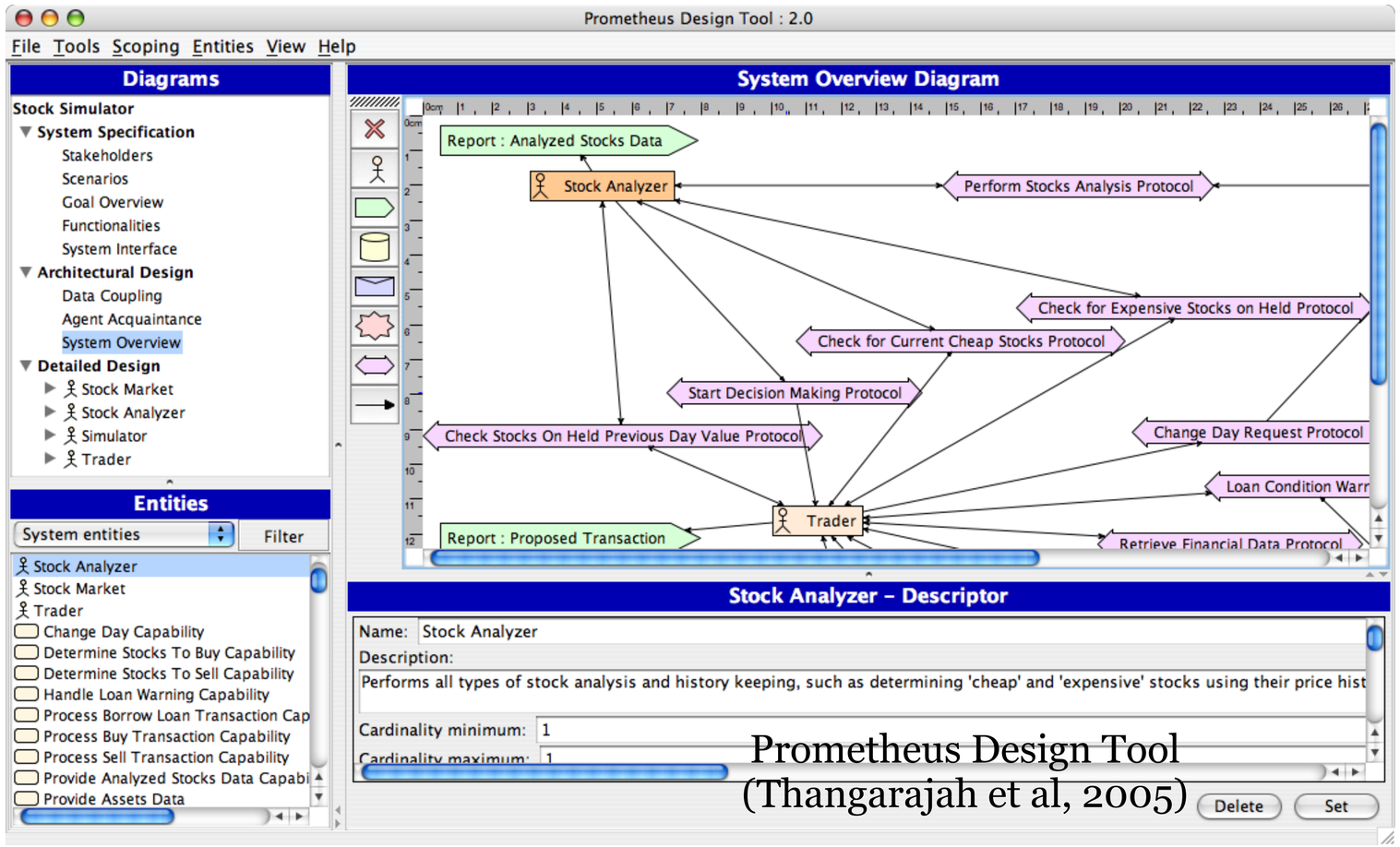
Techniques

- Example: Define agent types by grouping smaller chunks, considering coupling and cohesion
- Example: Scenarios → Interaction Diagrams → Protocols

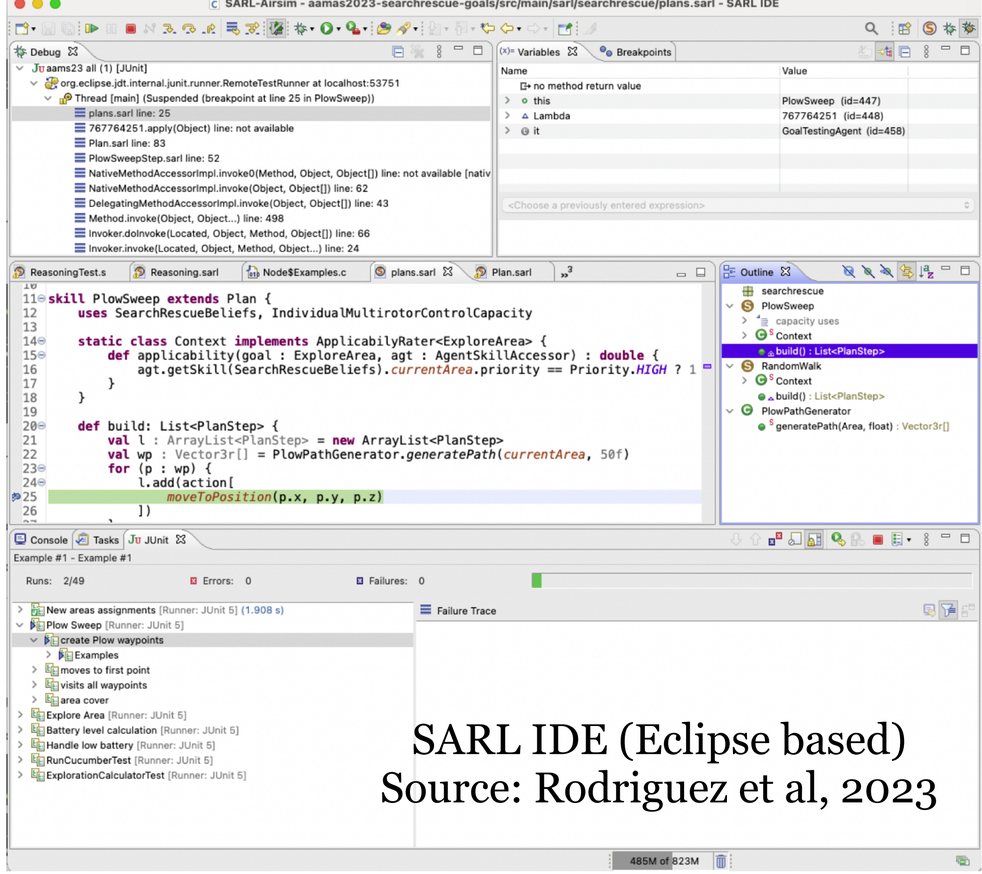


Tools

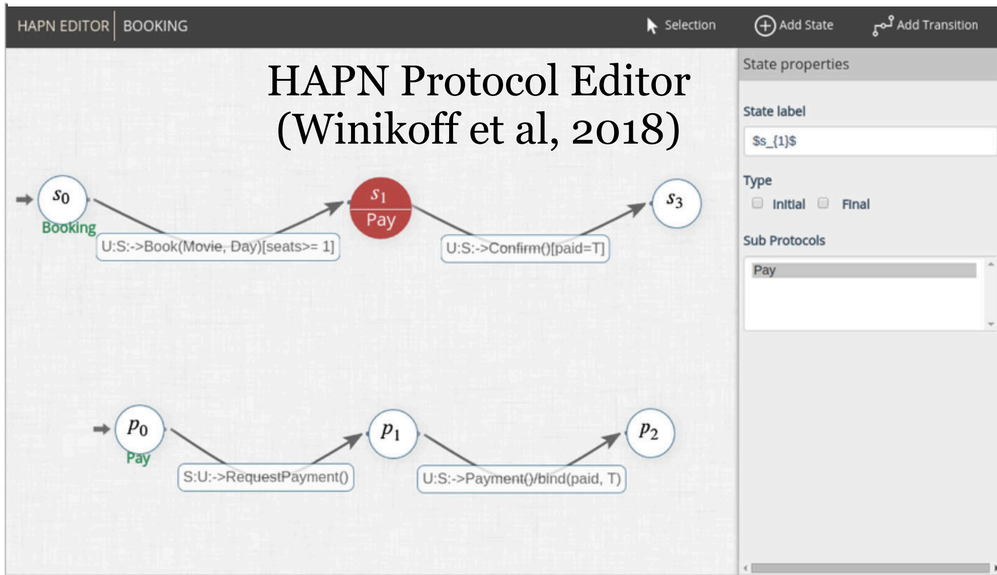
- Support design, coding, testing, debugging, ...



Prometheus Design Tool
(Thangarajah et al, 2005)



SARL IDE (Eclipse based)
Source: Rodriguez et al, 2023



HAPN Protocol Editor
(Winikoff et al, 2018)

John Thangarajah, Lin Padgham, and Michael Winikoff. 2005. Prometheus design tool. AAMAS. https://doi.org/bhchsb
Sebastian Rodriguez, John Thangarajah, and Michael Winikoff. 2023. A Behaviour-Driven Approach for Testing Requirements via User and System Stories in Agent Systems. AAMAS. https://www.ifaamas.org/Proceedings/aamas2023/pdfs/p1182.pdf
Michael Winikoff, Nitin Yadav, and Lin Padgham. 2018. A new Hierarchical Agent Protocol Notation. JAAMAS. https://doi.org/gct7gn

Languages

- ***Could have a whole talk just on this!***
- Many languages (e.g. 2APL, 3APL, AgentSpeak, Jason, JaCaMo [Jason+CArtAgO+Moise], Astra, Brahms, GOAL, GOLOG, ConGolog, IndiGolog, Gwendolen, JACK, JADE, Jadescript, Jadex, JIAC, MetateM, SARL ...) — Bordini et al, 2005 & 2009; Bordini & Dix, 2016
- BDI (Belief-Desire-Intention) common (next slide ... will show an example AgentSpeak program later) — Bratman, 1987; Ingrand et al, 1992

Rafael H. Bordini, Mehdi Dastani, Jürgen Dix, and Amal El Fallah Seghrouchni. 2005. Multi-Agent Programming: Languages, Platforms and Applications. Multiagent Systems, Artificial Societies, and Simulated Organizations 15, Springer, ISBN 0-387-24568-5

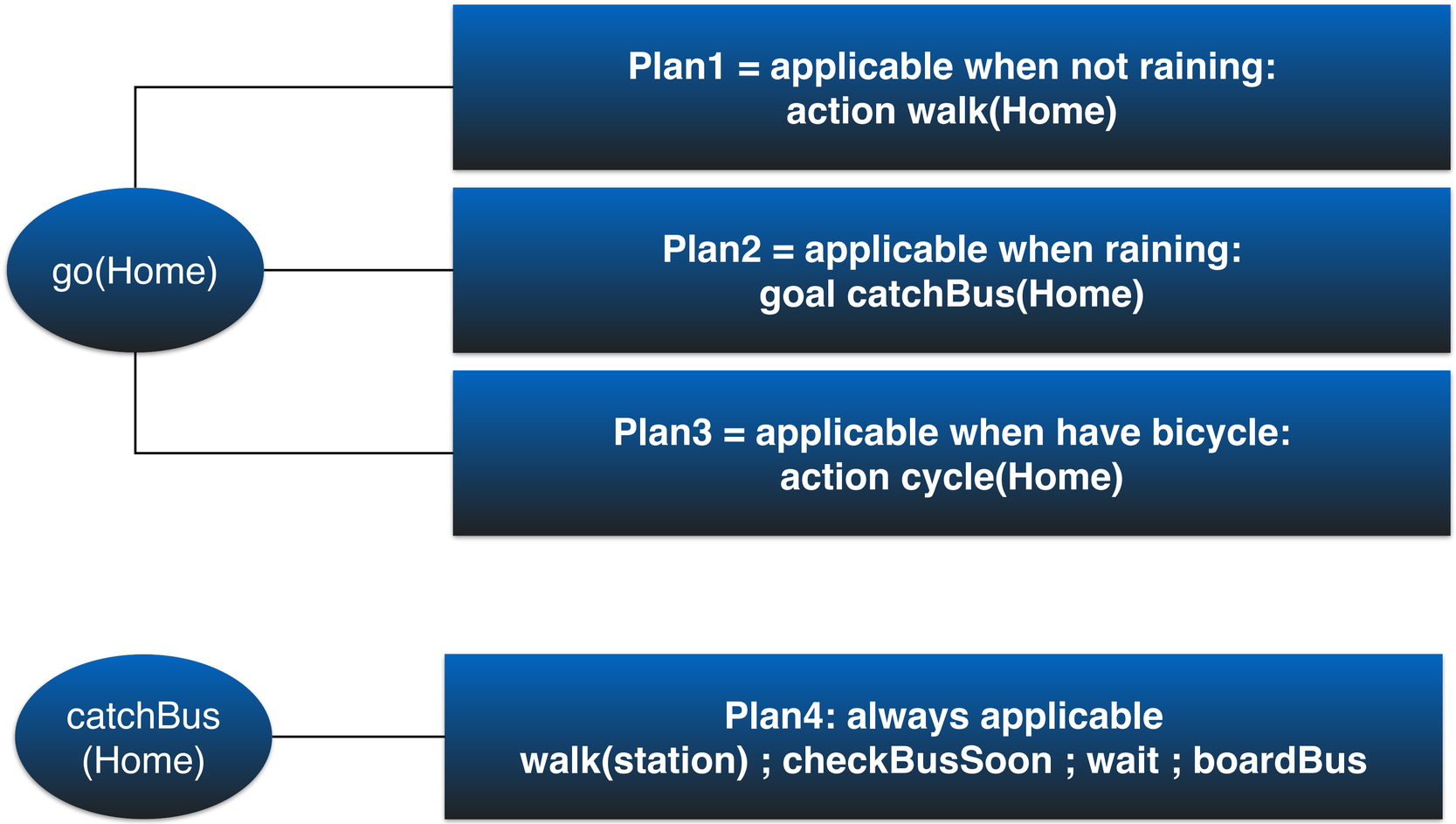
Rafael H. Bordini, Mehdi Dastani, Jürgen Dix, and Amal El Fallah Seghrouchni. 2009. Multi-Agent Programming, Languages, Tools and Applications. Springer, ISBN 978-0-387-89298-6

Rafael H. Bordini and Jürgen Dix. 2016. Programming Multiagent Systems, chapter 13 of Multiagent Systems, Second Edition Edited by Gerhard Weiss, 2016. ISBN: 9780262533874. MIT Press. Pages 587-639

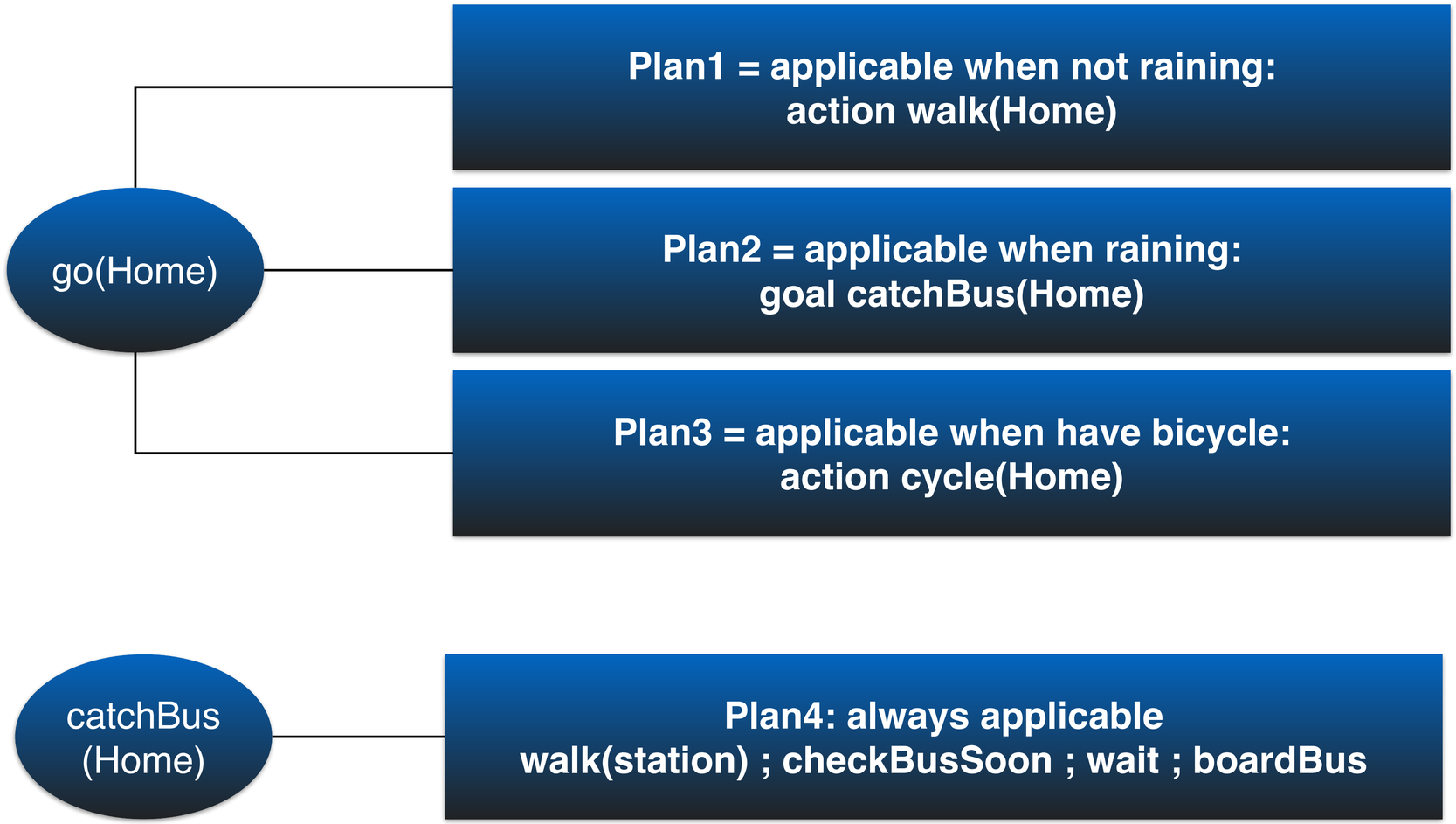
Michael E. Bratman. 1987. Intentions, Plans, and Practical Reason. Harvard University Press, Cambridge, MA.

F. F. Ingrand, M. P. Georgeff and A. S. Rao. 1993. An architecture for real-time reasoning and system control. IEEE Expert. http://doi.org/dz574s

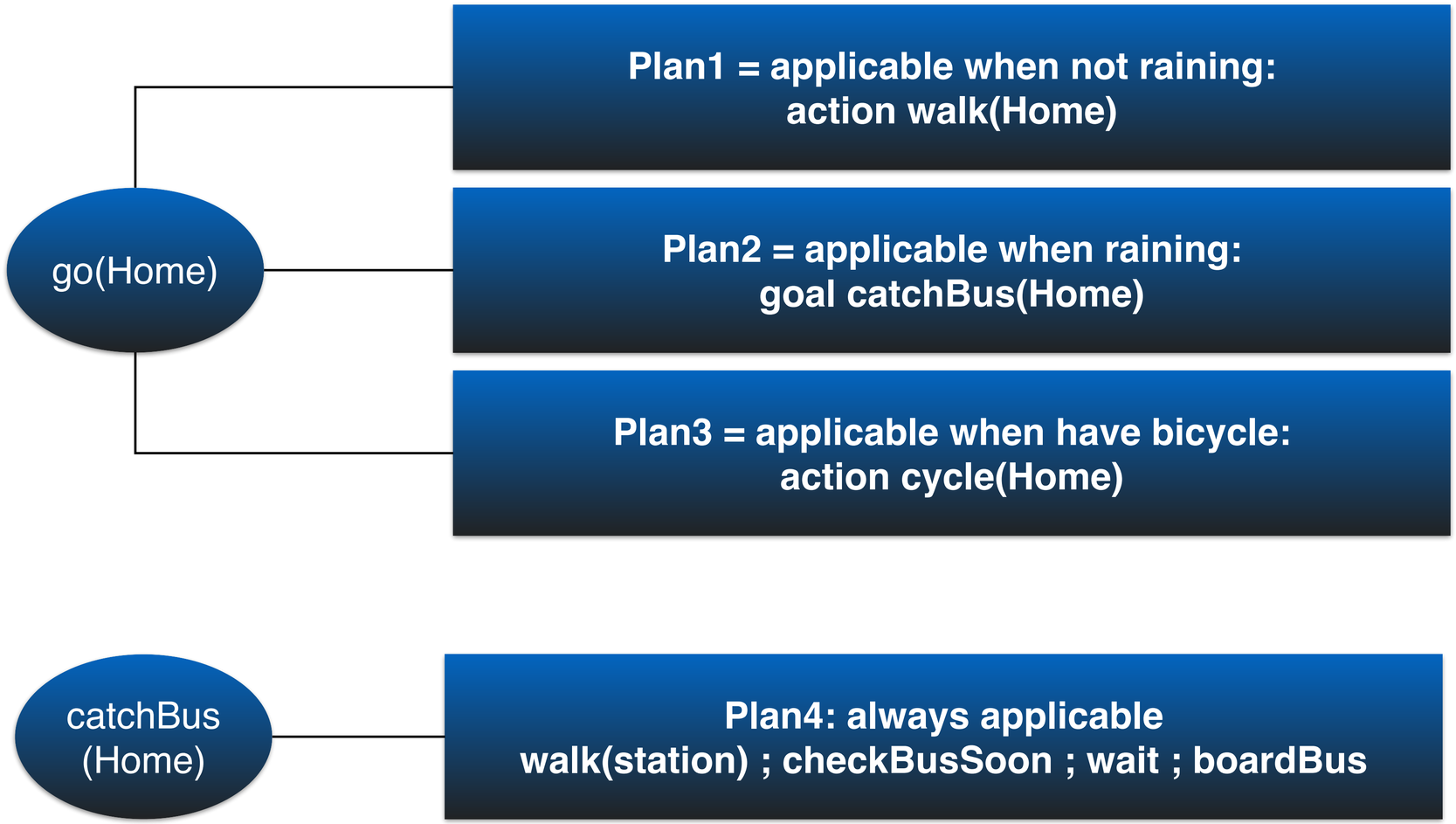
BDI: Goals & Plans



BDI: Goals & Plans

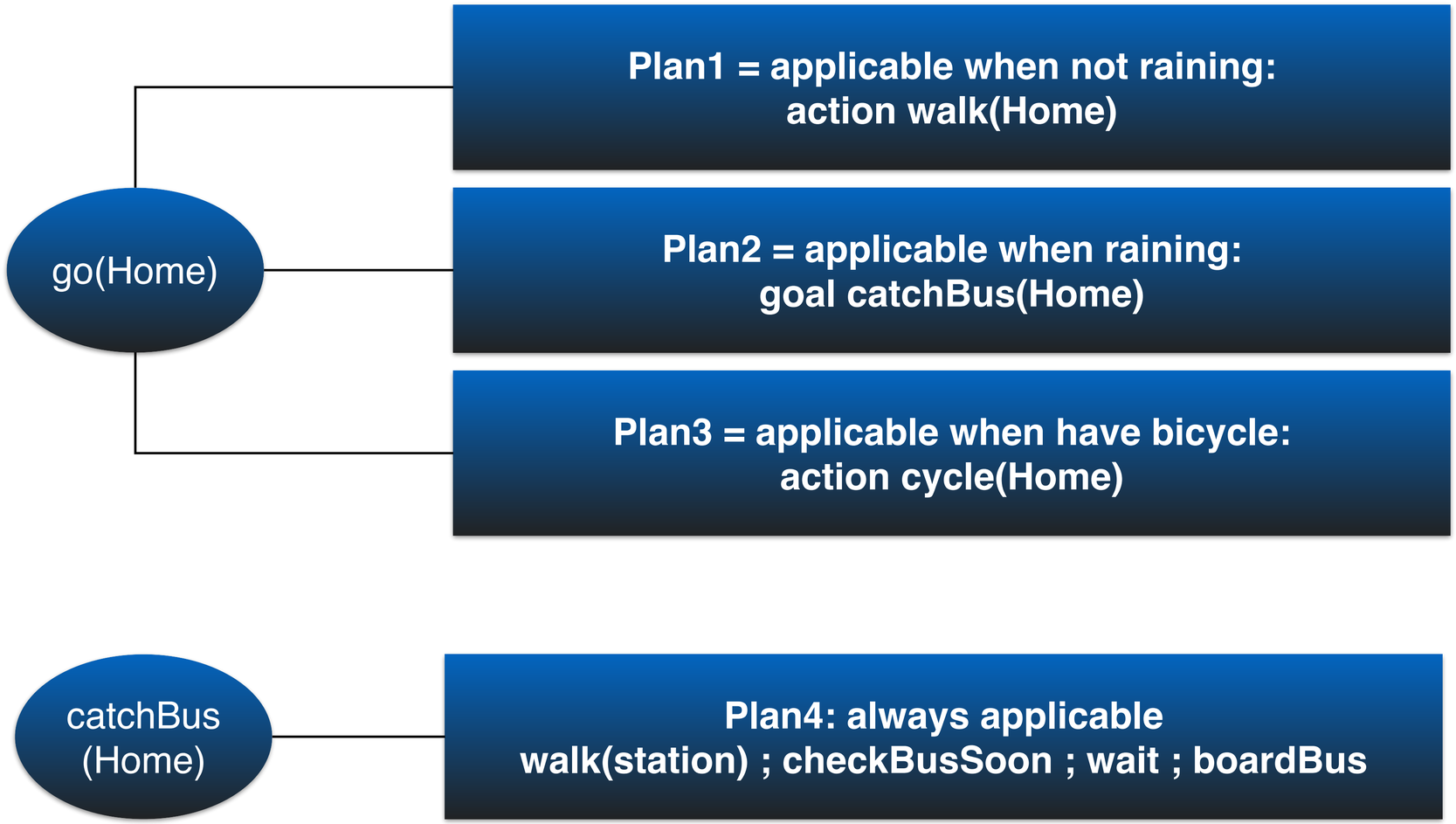


BDI: Goals & Plans



Goals and plans provide a compact way of expressing many possible behaviours

BDI: Goals & Plans



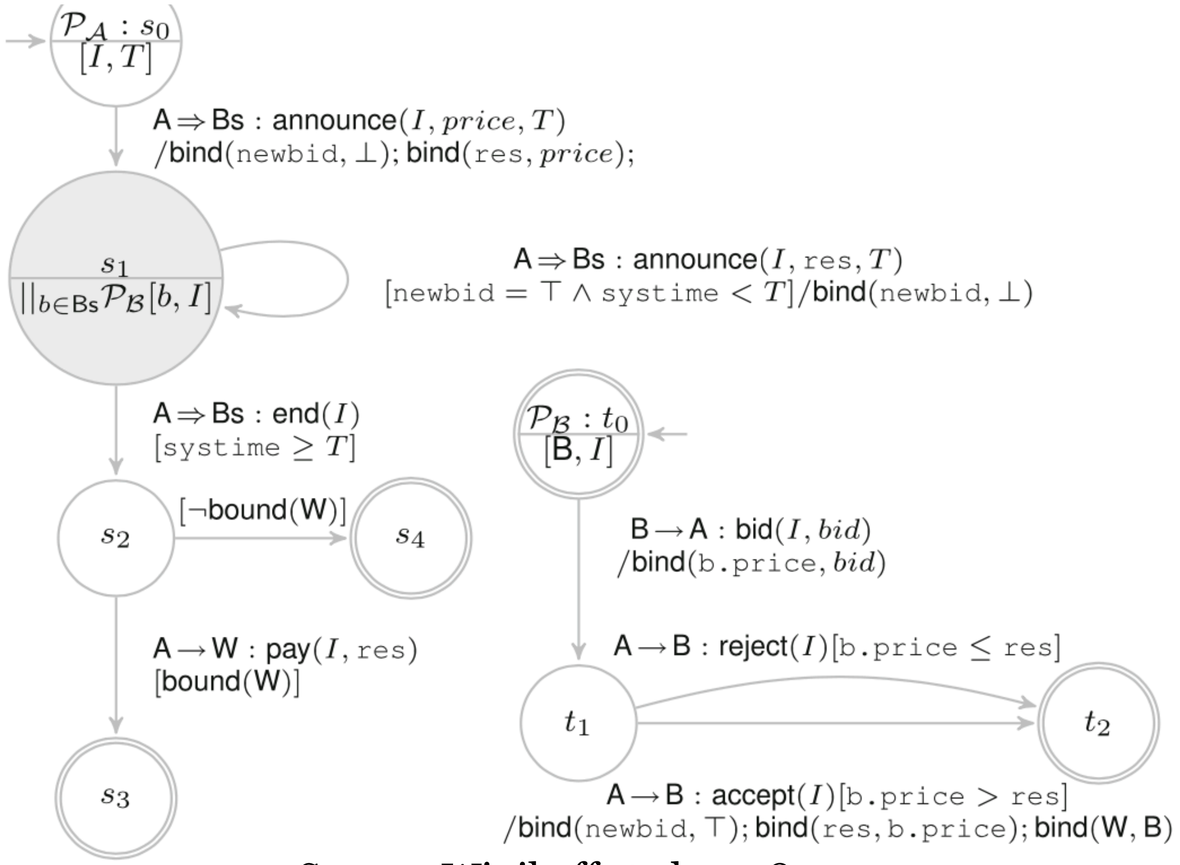
Goals and plans provide a compact way of expressing many possible behaviours

Goals also play a role in design and in agent reasoning

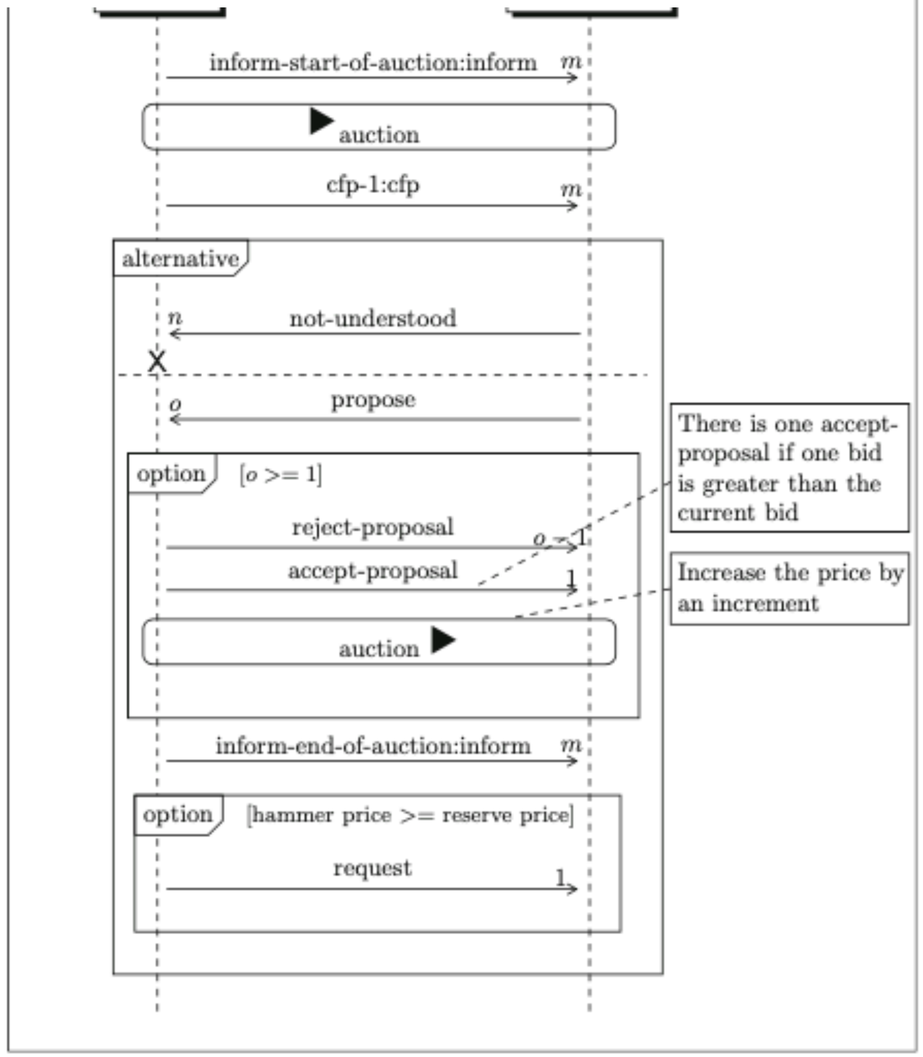
Some key EMAS topics

- Flexible interactions
- Assurance & debugging
- ...

Flexible Interactions: Beyond message-centric protocols



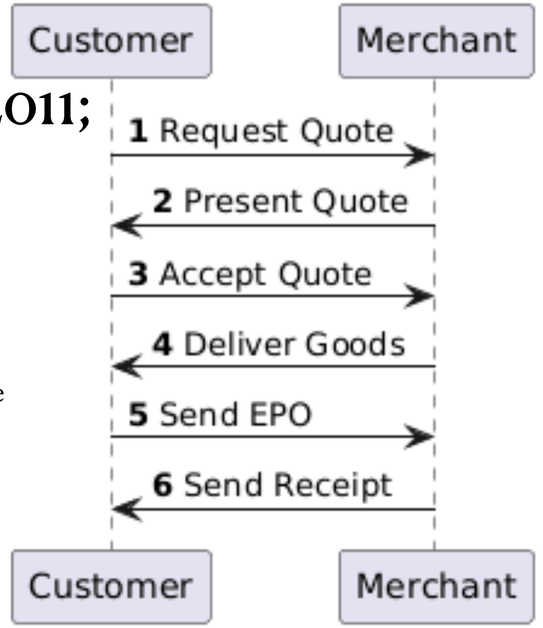
Source: Winikoff et al, 2018



Michael Winikoff, Nitin Yadav, and Lin Padgham. 2018. A new Hierarchical Agent Protocol Notation. Journal of Autonomous Agents and Multi-Agent Systems. https://doi.org/gct7gn

Flexible interactions …

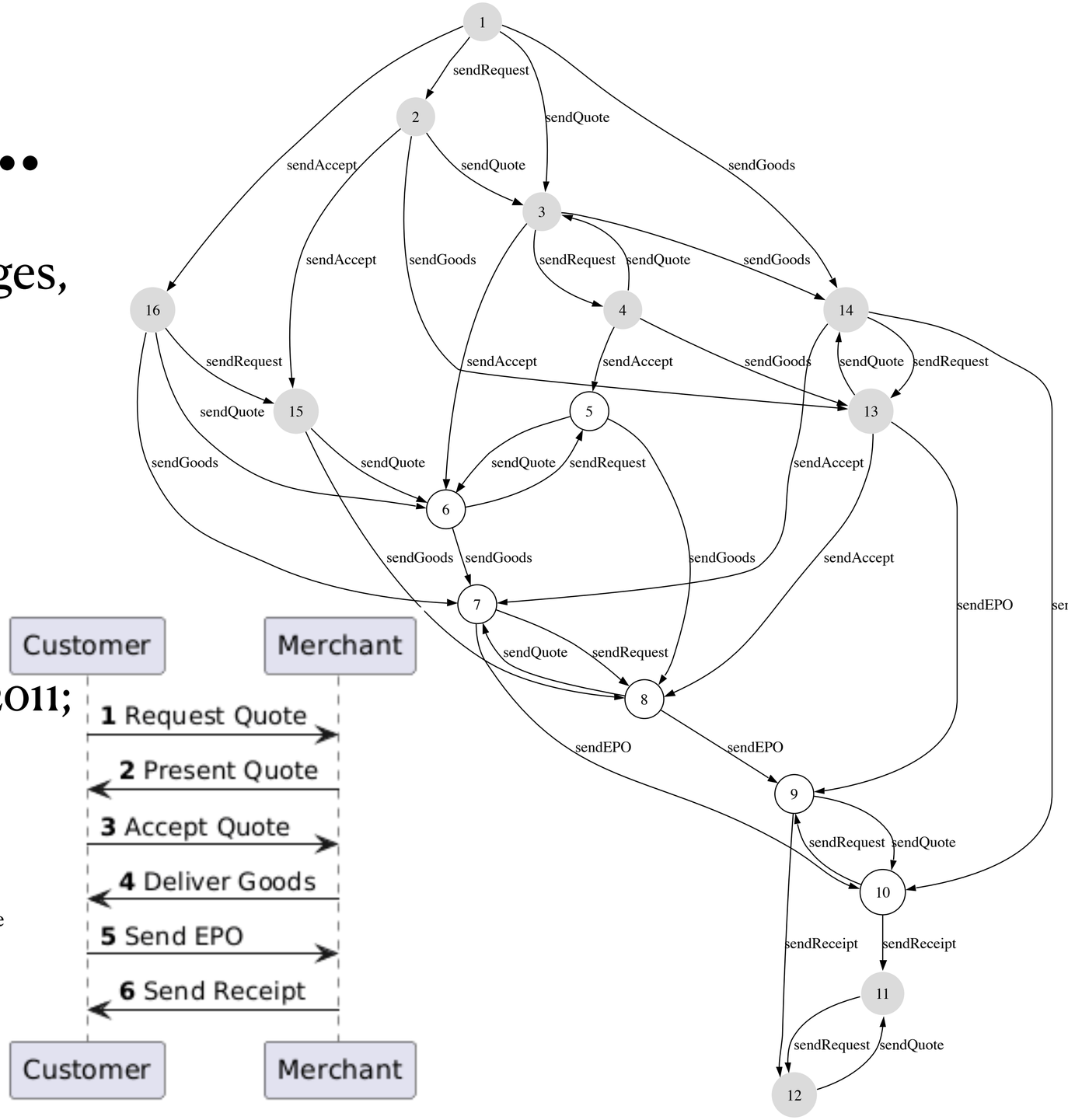
- Interaction protocols focus on messages, and can be too rigid
- Example (right): six-step process with some constraints
- Alternative approaches: social commitments (Yolum & Singh, 2002), information-based protocols (Singh, 2011; Chopra & Christie V, 2023)



Pınar Yolum and Munindar P. Singh. 2002. Commitment Machines. ATAL. https://doi.org/c8d6cx
Munindar P. Singh. 2011. Information-driven interaction-oriented programming: BSPL, the blindingly simple protocol language. AAMAS. https://www.ifaamas.org/Proceedings/aamas2011/papers/A4_B57.pdf
Amit K. Chopra and Samuel H. Christie V. 2023. Communication Meaning: Foundations and Directions for Systems Research. AAMAS. https://www.ifaamas.org/Proceedings/aamas2023/pdfs/p1786.pdf

Flexible interactions …

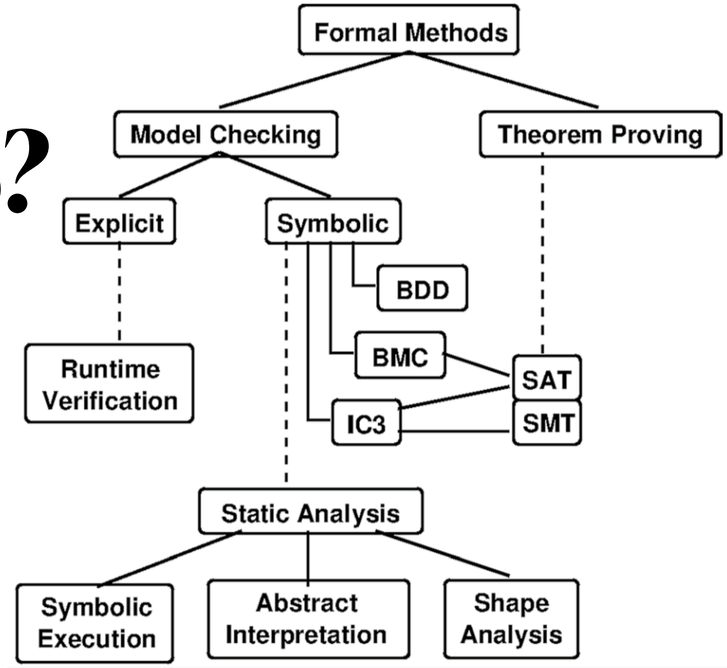
- Interaction protocols focus on messages, and can be too rigid
- Example (right): six-step process with some constraints
- Alternative approaches: social commitments (Yolum & Singh, 2002), information-based protocols (Singh, 2011; Chopra & Christie V, 2023)



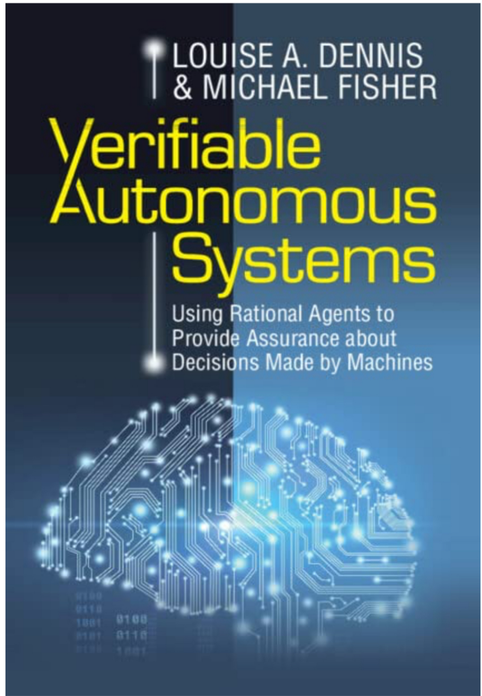
Pınar Yolum and Munindar P. Singh. 2002. Commitment Machines. ATAL. https://doi.org/c8d6cx
Munindar P. Singh. 2011. Information-driven interaction-oriented programming: BSPL, the blindingly simple protocol language. AAMAS. https://www.ifaamas.org/Proceedings/aamas2011/papers/A4_B57.pdf
Amit K. Chopra and Samuel H. Christie V. 2023. Communication Meaning: Foundations and Directions for Systems Research. AAMAS. https://www.ifaamas.org/Proceedings/aamas2023/pdfs/p1786.pdf

How can I *verify* my MAS?

- Range of techniques …
- Formal verification of agent programs (Dennis & Fisher, 2023)



Source: Fisher et al, 2021



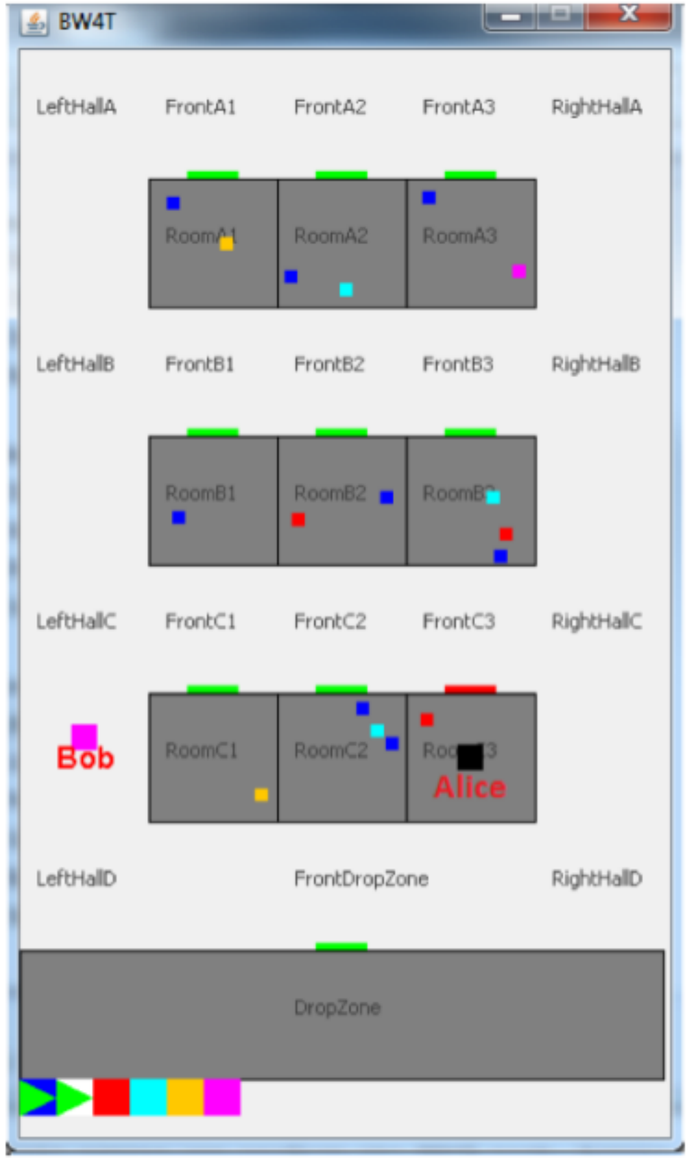
Louise Dennis and Michael Fisher. 2023. Verifiable Autonomous Systems: Using Rational Agents to Provide Assurance about Decisions Made by Machines. Cambridge University Press, 2023. ISBN 978-1108484992

Michael Fisher, Viviana Mascardi, Kristin Yvonne Rozier, Bernd-Holger Schlingloff, Michael Winikoff, and Neil Yorke-Smith. 2021. Towards a framework for certification of reliable autonomous systems. JAAMAS. https://doi.org/ms98

Winikoff, 2017

How can I *debug* my MAS?

- Agent programs hard to debug: parallelism, complex control flow
- Key idea: use interactive questioning (“Why?”) (Hindriks 2012, Ko & Myers 2008)
- Example: Blocks World for Teams (BW4T)
- Complementary to algorithmic debugging (Ahlbrecht, 2023)

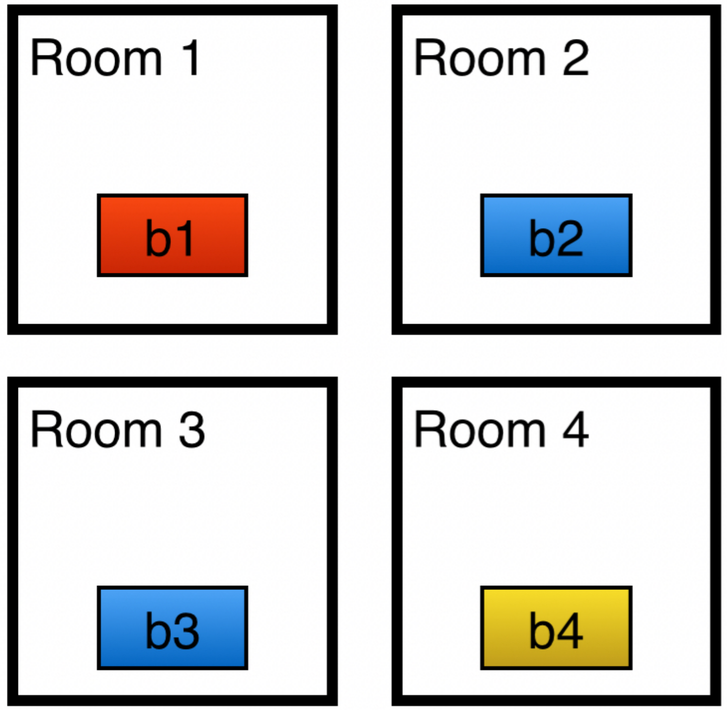


Michael Winikoff. 2017. Debugging Agent Programs with Why? Questions. AAMAS. https://www.ifaamas.org/Proceedings/aamas2017/pdfs/p251.pdf

Koen V. Hindriks. 2012. Debugging Is Explaining. PRIMA https://doi.org/ms99

Amy J. Ko and Brad A. Myers. 2008. Debugging reinvented: asking and answering why and why not questions about program behavior. ICSE. https://doi.org/dgvqm8

Tobias Ahlbrecht. 2023. An algorithmic debugging approach for belief-desire-intention agents. Ann Math Artif Intell. https://doi.org/ms96




```
1   goto(room1)
2   gotoBlock(b1) % red
3   pickUp
4   goto(dropzone)
5   putDown % now need yellow ...
6   goto(room2) % exploring ...
7   goto(room3)
8   goto(room4)
9   gotoBlock(b4) % yellow
10  pickUp
11  goto(dropzone)
12  putDown % now need blue ...
13  gotoBlock(b3) % blue
14  pickUp
15  call(deliver)
16  gotoBlock(b2) % blue
17  pickUp % FAILED (because already holding b3)
18  goto(dropzone)
19  putDown % now done
```

+!deliver : nextColour(done) ← +done. *% If done then stop*
% select a block of the right colour and go get and deliver it
+!deliver : colour(B, C) ∧ nextColour(C) ∧ ¬holding(B) ← gotoBlock(B) ; pickUp ; !deliver.
+!deliver : holding(B) ∧ colour(B,C) ∧ nextColour(C) ← goto(dropzone) ; putDown ; !deliver.
% if holding a block that is not the next colour required then put it down (this may occur if e.g. someone else delivers a block, so the next colour changes)
+!deliver : holding(B) ∧ colour(B,C) ∧ ¬nextColour(C) ← putDown ; !deliver.
% if I know of a place that I've not yet visited then go there (explore)
+!deliver : place(P) ∧ ¬beenthere(P) ← goto(P) ; !deliver.

Debugging with Questions

Q1: why did pickUp at 17?

A1: because (i) !deliver at 15, (ii) plan π_2 has true context condition of "***colour(b2,blue) & nextColour(blue) & ~holding(b2)***", and (iii) the step gotoBlock(b2) succeeded

Q2: why believe colour(b2,blue) & nextColour(blue) & ~holding(b2) at 15?

A2: because putDown done at 12, and colour(b2,blue) & ***~holding(b2)*** was true

Q3: why believe ~holding(b2) at 12?

A3: that's been true since the beginning of execution ...

...
11 goto(dropzone)
Q3 12 putDown
13 gotoBlock(b3) *% blue*
14 pickUp
Q2 15 call(deliver)
16 gotoBlock(b2) *% blue*
Q1 17 pickUp *% FAILED*
...

π_2 = +!deliver :
colour(B, C)
& nextColour(C)
& ~holding(B)
← gotoBlock(B) ;
pickUp ; !deliver.

Summary (so far)

- EMAS is concerned with engineering agent systems
- EMAS encompasses concepts, methodologies (process, notation, techniques, tools), languages
- Some specific key work: goals, flexible interactions, assurance
- What about LLMs? (Next slides)

Link to these slides:
winikoff.github.io/pdf/DXC.pdf

What about LLMs?

- EMAS and LLMs: how can we use EMAS concepts to engineer LLM-based systems that are reliable & transparent?
- LLMs and EMAS: how can LLMs be used by (“traditional”) agent systems?
- How to bridge communities? (Role for standardisation and teaching)

“How can we engineer autonomous agents that use LLMs, but leverage EMAS concepts and techniques to improve their effectiveness, flexibility, reliability, and transparency?”

“How can we leverage Generative AI to lower the entry barrier to engineering autonomous agents, and to improve the agents’ adaptiveness, interoperability, and usability?

(Ciortea *et al.*)

Andrei Ciortea et al. (30 authors) Engineering Multi-Agent Systems and Generative AI: Report from the Agent Toolkits 2025 Community Session (under review)

Link to these slides:
winikoff.github.io/pdf/DXC.pdf

AAMAS: beyond EMAS …

See https://cyprusconferences.org/aamas2026/call-for-papers-main-track/ for descriptions

- Learning and Adaptation (LEARN)
- Generative and Agentic AI (GAAI) **NEW**
- Game Theory and Economic Paradigms (GTEP)
- Coordination, Organizations, Institutions, Norms, and Ethics (COINE)
- Search, Optimization, Planning, and Scheduling (SOPS)
- Representation, and Reasoning (RR)
- Engineering and Analysis of Multiagent Systems (EMAS)
- Modeling and Simulation of Societies (SIM)
- Human-Agent Interaction (HAI)
- Robotics and Control (ROBOT)
- Innovative Applications (IA)

Link to these slides:
winikoff.github.io/pdf/DXC.pdf

Broader relevance to LLMs …

AAMAS Concept	Description	Agentic AI?	Role in Agentic AI
BDI Architecture	Explicit representation of beliefs, desires, intentions	❌ Not explicit	Enables interpretable, goal-driven behavior; essential for reasoning and alignment
Communication Protocols	Structured inter-agent messages with semantics	⚠️ Ad-hoc, via natural language	Provides reliable coordination and prevents miscommunication in multi-agent systems
Mechanism Design / Incentive Alignment	Systems design for self-interested agents to act in desirable ways	❌ Rarely applied	Aligns agent behaviors with system-wide objectives; avoids unintended or harmful outcomes
Multi-Agent Planning / Coordination	Joint plans, task decomposition, distributed execution	⚠️ Shallow coordination	Supports efficient task distribution and cooperation among agents
Negotiation / Argumentation	Agents negotiate over goals, actions, resources	❌ Absent	Enables dynamic conflict resolution and adaptive collaboration
Norms, Institutions, Roles	Agents governed by social norms or institutional rules	❌ Not modeled	Enforces behavioral constraints and fosters trust and stability in open environments
Trust / Reputation	Evaluating agent reliability over time	❌ Absent	Critical for long-term collaboration and risk-aware decision making
Social Choice / Voting	Group decision-making among agents	❌ Not used	Enables collective intelligence and democratic coordination in multi-agent scenarios
Game Theory (Co-op/Adversarial)	Strategic reasoning about others' choices	⚠️ Minimal use	Improves agents' ability to operate under uncertainty and strategic interaction
Theory of Mind / Belief Modeling	Reasoning about other agents' beliefs, goals, intentions	⚠️ Very rudimentary	Allows modeling and anticipation of others' behavior—key for safety, alignment, and cooperation

Figure 1 from *Agentifying Agentic AI.* Virginia Dignum & Frank Dignum. 2025. https://arxiv.org/abs/2511.17332

Summary

- EMAS is concerned with engineering agent systems
- EMAS encompasses concepts, methodologies (process, notation, techniques, tools), languages
- Some specific key work: goals, flexible interactions, assurance
- What about LLMs?
  - EMAS can contribute to engineering Agentic systems
  - LLMs can be used in “traditional” agent systems
- Beyond EMAS: AAMAS has more to offer …

Link to these slides:
winikoff.github.io/pdf/DXC.pdf