Tick those statements you believe are true:

- [ ] People only plan when they have to because the benefit of an optimal plan does not always justify the effort of planning.
- [ ] For humans, planning is a subconscious process, which is why computational planning is so hard.
- [ ] Planning involves a mental simulation of actions to foresee future world states and compare them to goals.
- [ ] In Artificial Intelligence, planning is only concerned with the search for computationally optimal plans.
- [ ] Domain-specific planning is used when efficiency is vital, whereas domain-independent planning is good for planning from first principles.

Submit

Skip

On one bank of a river are three missionaries (black triangles) and three cannibals (red circles). There is one boat available that can hold up to two people and that they would like to use to cross the river. If the cannibals ever outnumber the missionaries on either of the river's banks, the missionaries will get eaten. How can the boat be used to safely carry all the missionaries and cannibals across the river?

Try to define the Missionaries and Cannibals problem as a state-transition system. That is, define what the possible world states are, what the actions and events are, and what the state-transition function looks like (best done by drawing the state-transition graph). If you've done all this, it should be easy to see the solution to the problem.

How long is the solution to this problem, that is, how many actions are necessary to get from the initial state to the goal state. Type the number of actions necessary in the box provided:

Submit

Skip

The following statements are either true or false. Tick the box for those statements that are true.

- [ ] State-transition systems can represent actions that occur in parallel.
- [ ] A state-transition graph may have multiple outgoing and/or incoming (directed) edges that are labelled with the same action.
- [ ] A plan consists of a set of actions that are organized into some structure and a plan always achieves some objective.
- [ ] During plan execution, we can use the observation function to determine which state exactly the system is in.
- [ ] In dynamic planning, one can always try to re-plan, that is, find a plan that achieves the original objective, but from the current state.

Submit

Skip

Tick those statements that you believe to be true:

- [ ] Both state-transition systems and search problems consist of exactly four components; therefore they must fundamentally be the same.
- [ ] Search problems include an initial state and goal states, whereas state-transition systems simply describe the way in which the world may evolve.
- [ ] In general, state-transition systems allow for non-determinsim. Search problems must be deterministic.
- [ ] The definition of a state-transition system requires the set of possible states to be enumerable. There is no such constraint on search problems.

Submit

Skip

Tick those statements that you believe to be true:

- [ ] The search problem for the Missionaries and Cannibals problem is finite and discrete.
- [ ] The search problem for the Missionaries and Cannibals problem is fully observable.
- [ ] The search problem for the Missionaries and Cannibals problem is deterministic.
- [ ] The search problem for the Missionaries and Cannibals problem is static.

Submit

Skip

On one bank of a river are three missionaries (black triangles) and three cannibals (red circles). There is one boat available that can hold up to two people and that they would like to use to cross the river. If the cannibals ever outnumber the missionaries on either of the river's banks, the missionaries will get eaten. How can the boat be used to safely carry all the missionaries and cannibals across the river?

Try to implement the general search algorithm just described. You can use LIFO and FIFO as queuing strategies to determine the order in which nodes are explored. These two strategies are known as depth-first and breadth-first search respectively. Be careful, depth-first search may descend down infinite branches, so best implement a depth cut-off. Then, extend your implementation with a hash table that stores all the nodes found so far. Print out a trace of the states the algorithm finds (in the order they are discovered) and see how much of the search space each algorithm explores.

Continue

Course Reading

- **Review of AI Planners to 1990**
- Hendler, J.A., Tate, A. and Drummond, M. (1990) "AI Planning: Systems and Techniques", AI Magazine Vol. 11, No. 2, pp.61-77, Summer 1990, AAAI Press.
- http://aaaipress.org/ojs/index.php/aimagazine/article/download/833/751

- **Knowledge-Based Planners**
- Wilkins, D. E. and desJardins, M. (2001) "A Call for Knowledge-based Planning", AI Magazine, Vol. 22, No. 1, pp. 99-115, Spring 2001, AAAI Press.
- http://www.aaai.org/ojs/index.php/aimagazine/article/view/1547/ or http://www.ai.sri.com/pub_list/808

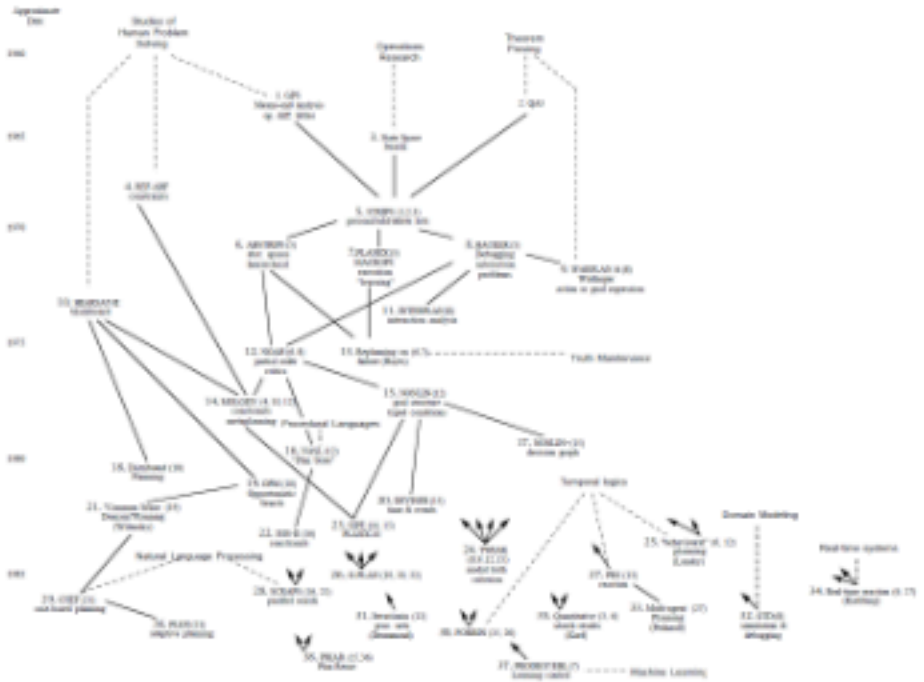
Course Readings Wiki page

https://share.coursera.org/wiki/index.php/AIPLAN:Readings

Continue

You can view a larger version of this image by clicking here now or at any time later via the lecture resources...

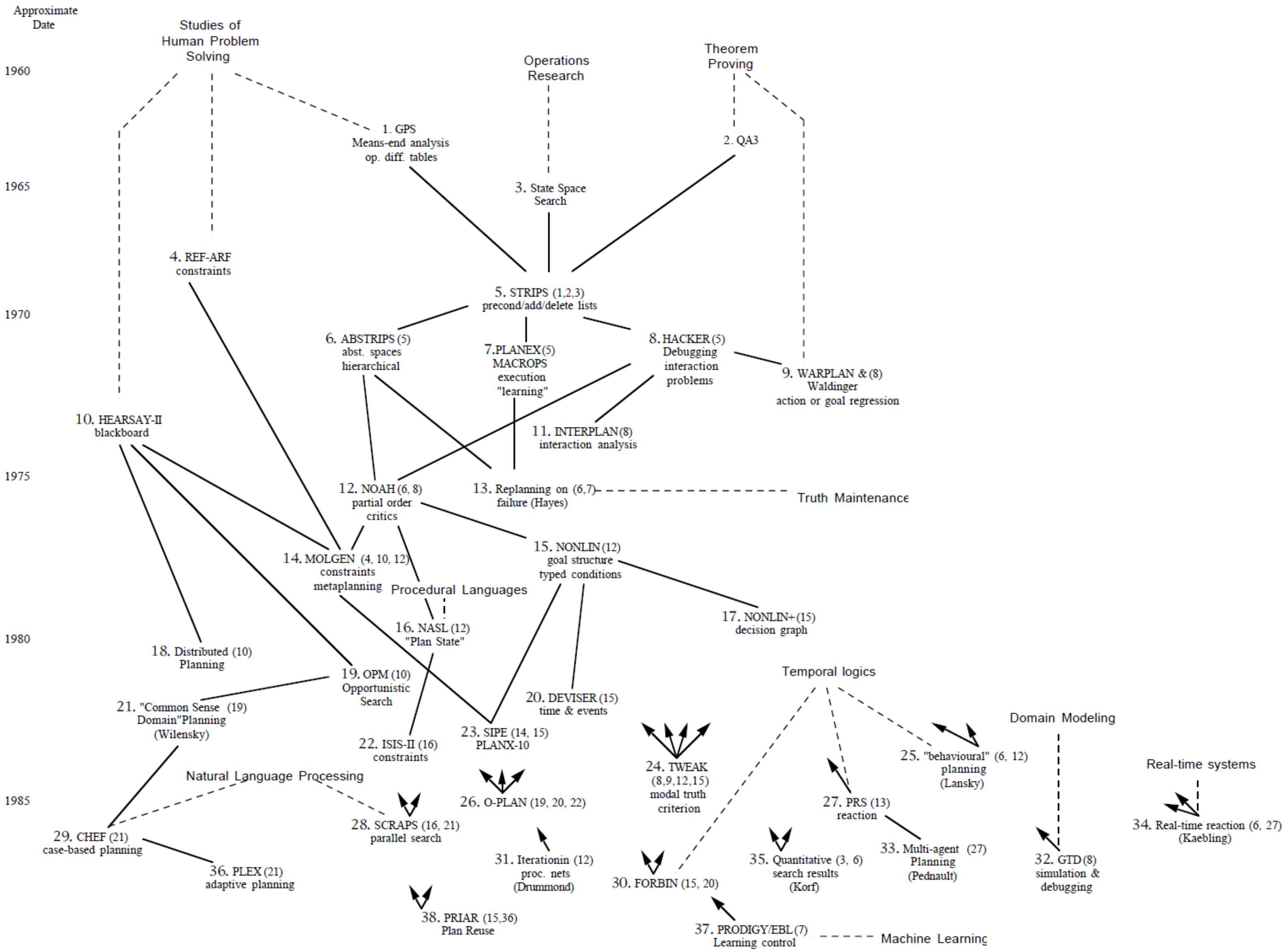
PNG Image



Note the origins of early AI planning work in:

1. Studies of Human Problem Solving
2. Operations Research
3. Theorem Proving

Continue



Approximate Date
1960
1965
1970
1975
1980
1985
Studies of Human Problem Solving
Operations Research
Theorem Proving
1. GPS Means-end analysis op. diff. tables
2. QA3
3. State Space Search
4. REF-ARF constraints
5. STRIPS (1,2,3) precond/add/delete lists
6. ABSTRIPS (5) abst. spaces hierarchical
7.PLANEX (5) MACROPS execution "learning"
8. HACKER (5) Debugging interaction problems
9. WARPLAN & (8) Waldinger action or goal regression
10. HEARSAY-II blackboard
11. INTERPLAN (8) interaction analysis
12. NOAH (6, 8) partial order critics
13. Replanning on (6,7) failure (Hayes)
Truth Maintenance
14. MOLGEN (4, 10, 12) constraints metaplanning
15. NONLIN (12) goal structure typed conditions
Procedural Languages
16. NASL (12) "Plan State"
17. NONLIN+ (15) decision graph
18. Distributed (10) Planning
19. OPM (10) Opportunistic Search
20. DEVISER (15) time & events
Temporal logics
21. "Common Sense (19) Domain"Planning (Wilensky)
22. ISIS-II (16) constraints
23. SIPE (14, 15) PLANX-10
24. TWEAK (8,9,12,15) modal truth criterion
25. "behavioural" (6, 12) planning (Lansky)
Domain Modeling
Real-time systems
Natural Language Processing
26. O-PLAN (19, 20, 22)
27. PRS (13) reaction
28. SCRAPS (16, 21) parallel search
29. CHEF (21) case-based planning
30. FORBIN (15, 20)
31. Iterationin (12) proc. nets (Drummond)
32. GTD (8) simulation & debugging
33. Multi-agent (27) Planning (Pednault)
34. Real-time reaction (6, 27) (Kaebling)
35. Quantitative (3, 6) search results (Korf)
36. PLEX (21) adaptive planning
37. PRODIGY/EBL (7) Learning control
Machine Learning
38. PRIAR (15,36) Plan Reuse

Video demonstrations of a couple of sample applications of I-X in the context of multiagent systems are available as resources attached to this video lecture.

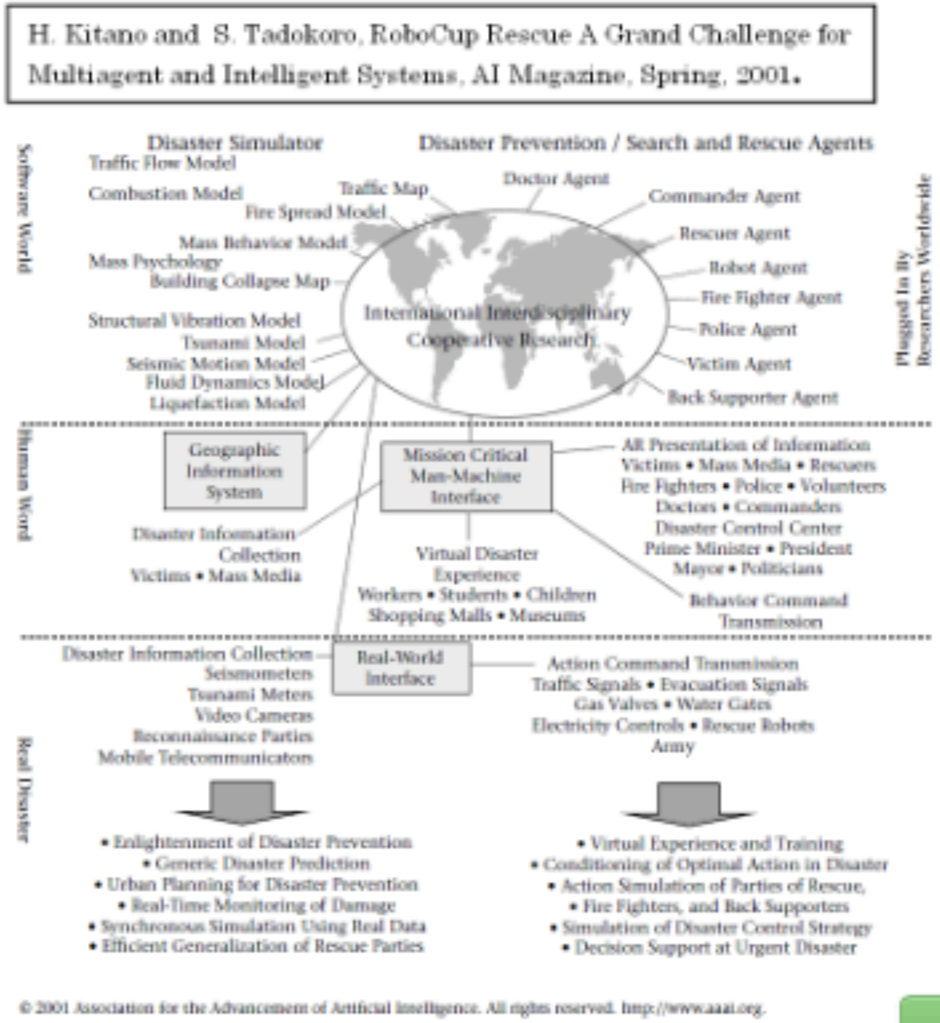
The videos are also available in the "Additional Stuff" section if you want to return to them at any time.

Continue

Figure from H. Kitano and S. Tadokoro, RoboCup Rescue A Grand Challenge for Multiagent and Intelligent Systems, AI Magazine, Spring, 2001.

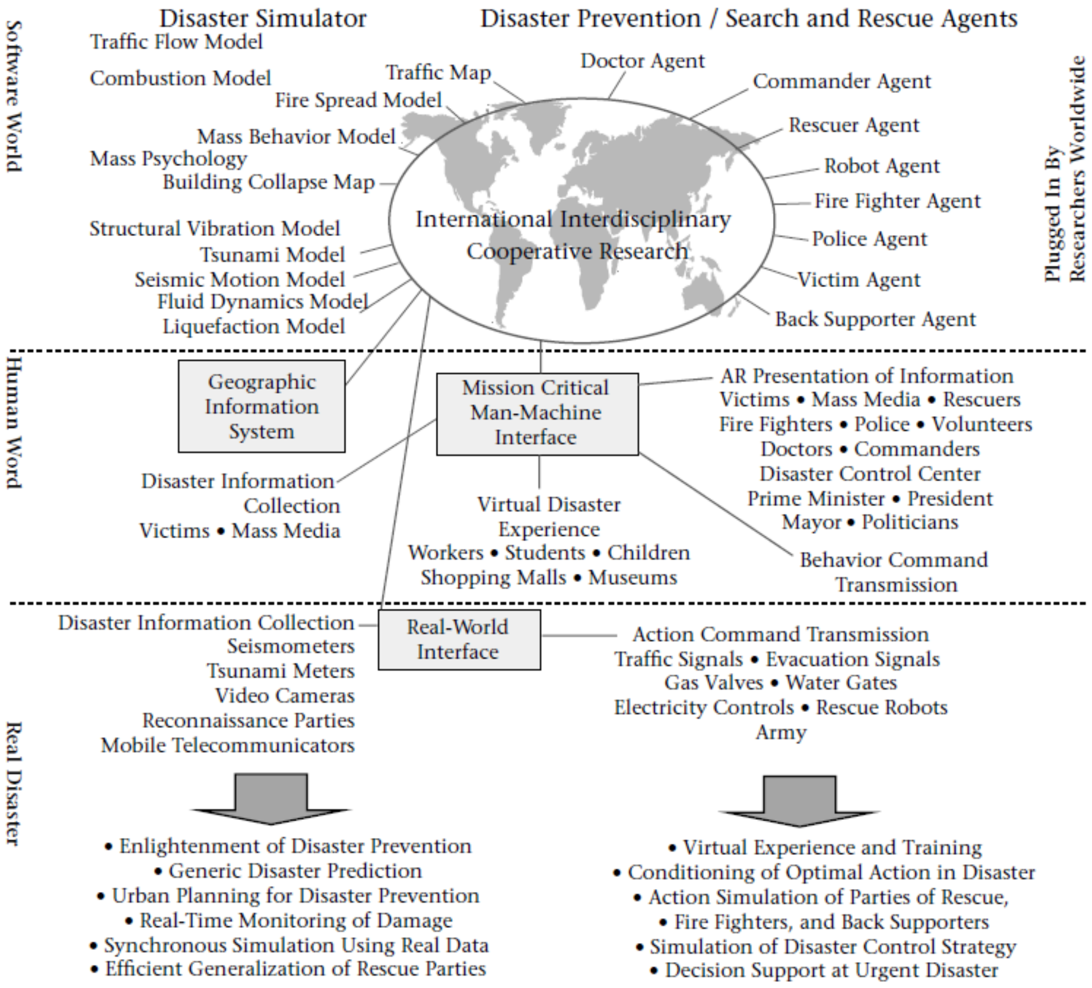
You can view a larger version of this image by clicking here now or at any time later via the lecture resources...

PNG Image



Continue

H. Kitano and S. Tadokoro, RoboCup Rescue A Grand Challenge for Multiagent and Intelligent Systems, AI Magazine, Spring, 2001.



A Video demonstration of O-Plan generating Unix systems administration scripts is available as a resource attached to this video lecture.

The video is also available in the "Additional Stuff" section if you want to return to it at any time.

Continue