The ADEPT Framework for Intelligent Autonomy

Dr. Michael Ricard / Dr. Stephan Kolitz
The Charles Stark Draper Laboratory, Inc.
555 Technology Square
Cambridge, MA 02139 USA

Abstract

This paper describes the design and implementation of Draper Laboratory's **All-Domain Execution and Planning Technology** (ADEPT) architecture for *intelligent autonomy*. Intelligent autonomy is the ability to plan and execute complex activities in a manner that provides rapid, effective response to stochastic and dynamic mission events. Thus, intelligent autonomy enables the high-level reasoning and adaptive behavior for an unmanned vehicle that is provided by an operator in man-in-the-loop systems.

Draper's intelligent autonomy has architecture evolved over a decade and a half beginning in the mid 1980's [3, 4, 6 and 12] culminating in an operational experiment funded under DARPA's Autonomous Minehunting and Mapping Technologies (AMMT) unmanned undersea vehicle program [15]. ADEPT continues to be refined through its application to current programs that involve air vehicles, satellites and higher-level planning used to direct multiple vehicles. The objective of ADEPT is to solidify a proven, dependable software approach that can be quickly applied to new vehicles and domains.

The architecture can be viewed as a hierarchical extension of the sense-think-act paradigm of intelligence and has strong parallels with the military's Observe-Orient-Decide-Act (OODA) loop [14]. The key elements of the architecture are planning and decision-making nodes comprising modules for situation assessment, plan generation, plan implementation and coordination. A reusable, object-oriented software framework has been developed that implements these functions. As the architecture is applied to new areas, only the application specific software needs to be developed.

This paper describes the core architecture in detail and discusses how this has been applied in the undersea, air, ground and space domains.

Introduction

In recent years, there has been a spectrum of development programs for air, space, ground, and underwater vehicles with the desire for increasing degrees of autonomy. Enabled by advances in intelligent autonomy, autonomous vehicles will ultimately be employed by the military as a force multiplier and as a means of reducing risk to military personnel and in nonmilitary applications in remote and hazardous locations, including search and rescue during a fire or natural disasters. For example, future missions of autonomous Uninhabited Combat Air Vehicles (UCAVs) may include lethal air to ground missions for suppression of enemy air defenses (SEAD), intelligence, surveillance, reconnaissance and targeting (ISRT), and logistics re-supply. To successfully carry out these classes of missions, UCAVs will require highly adaptive autonomous mission planning and control, real-time obstacle avoidance of both static and dynamic obstacles and path planning for high speed flight in complex terrain.

Here *Intelligent Autonomy* refers to a vehicle's capability of sensing its own state and the state of its environment, and, based on this situational awareness, of planning and executing actions that are designed to achieve specific objectives under defined constraints. Thus, intelligent autonomy requires the development of automated planning and control systems that plan the vehicle's mission prior to its deployment, control the vehicle's state during the execution of the planned mission and replan the mission in order to accommodate unanticipated events that may arise during mission execution.

Paper presented at the RTO AVT Course on "Intelligent Systems for Aeronautics",
held in Rhode-Saint-Genèse, Belgium, 13-17 May 2002, and published in RTO-EN-022.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **01 JUN 2003**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**
4. TITLE AND SUBTITLE **The ADEPT Framework for Intelligent Autonomy**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **The Charles Stark Draper Laboratory, Inc. 555 Technology Square Cambridge, MA 02139 USA**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM001519. RTO-EN-022, The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**12**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

As explained below, the challenges that intelligent autonomy must address are daunting: 1) developing and executing plans of activities that meet mission objectives and honor constraints, 2) dealing with uncertainty, and 3) providing a capability for dynamically adjusting a vehicle's plan in real time.

First, multiple (and often conflicting) objectives must be pursued in the face of a variety of both implicit and explicit constraints. Representative mission objectives for an Unmanned Combat Air Vehicle (UCAV) might include surveillance, reconnaissance, resupply, support and strike [10]. Implicit constraints are constraints that are imposed by the vehicle design (e.g., its fuel carrying capacity, weapon stores carrying capacity, performance envelope and subsystem capabilities). Explicit constraints are those that may be imposed by a higher planning authority including:

1. a required probability of survival or mission success,
2. time or ordering constraints that may be imposed on the pursuit of specific mission objectives,
3. navigation constraints, and
4. constraints imposed by the time available to plan.

Second, the fact that intelligent autonomy must accommodate the significant uncertainty in the system's knowledge of the current and future "state of the world" (i.e., state of the vehicle and its environment) also contributes to the complexity of the mission planning problem. Because of this uncertainty, the formulation of detailed plans far into the future or planning in advance for all possible contingencies is generally a futile exercise. As a consequence, the ability to replan onboard the mission is essential for autonomous vehicle applications, especially in poorly characterized environments where long endurance or a high probability of mission success is required.

The third challenge, to replan the mission in real time due to changes in the environment, commander's intent, changing opportunities, etc., adds another layer of complexity to the mission planning problem. This is due not only to the uncertainty with respect to the future state of the world that inevitably prevails at the time the mission is originally planned, but also to unforeseen, externally influenced events that can arise during the execution of the mission requiring a response in real time. For example, a higher planning authority may redefine the mission objectives or alter the constraints that are imposed on the pursuit of those objectives, the mission environment (threats, weather, etc.) might change, the vehicle might sustain failures or damage, or opportunities to accomplish additional objectives might arise.

Draper has designed the core ADEPT architecture to meet these intelligent autonomy challenges. Draper's ADEPT architecture is a hierarchical, closed-loop, real-time mission and trajectory planning capability for autonomous vehicles. This paper describes the development of ADEPT, and its application to a variety of domains. The paper is organized as follows. Section 2 discusses the functional components of the architecture and its roots from which it was developed, implemented and deployed in the highly successful DARPA Autonomous Minehunting and Mapping Technology (AMMT) Program. Section 3 describes the object–oriented design and implementation. Section 4 discusses the diverse domains employing this architecture and software.

Architecture Functional Description

ADEPT is based on Draper's core approach to autonomous mission planning systems [3]. The first fielded *operational implementation* of this core architecture was for DARPA's Advanced Minehunting and Mapping Technologies program. Additional capabilities have been introduced under both corporate sponsored research and externally funded programs. To make the solution of complex problems tractable, the problems are decomposed into simpler, decoupled subproblems that can be solved (nearly) independently.

Temporal hierarchical decompositions are often employed to simplify the implementation of the solution to real-time, closed-loop planning problems [4, 8]. These decompositions are characterized by higher levels that create plans with the greatest temporal scope (longest planning horizon) but with the least detail. At lower levels, the planning horizon becomes shorter (nearer term), but the level of detail of planned activities increases. The less detailed plans at the higher levels coordinate or guide the generation of solutions generated at the lower levels.

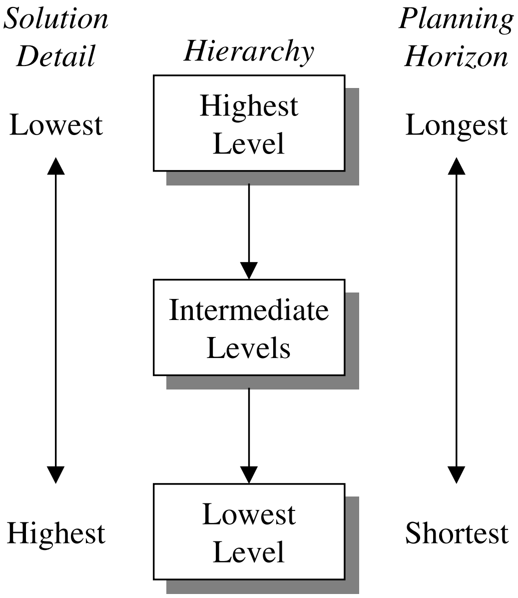

Figure 1: **Characteristics of Solutions at Various Levels of the Hierarchy**

Indeed, planning actions over extended periods of time at a high level of detail is *futile* because detailed actions planned on the basis of a specific prediction of the future may become obsolete well before they are to be executed due to an inability to accurately predict the future and *impractical* because the computational resources required to develop detailed plans for complex objectives over extended periods of time may be prohibitive.

Figure 2 shows the details of the functions performed at each level of the architecture in Figure 1. The key elements are modules for situation assessment, plan generation, plan implementation, and coordination. Figure 2 shows plan implementation providing input to the *system to be controlled*. Note that in the three tier hierarchical system shown in Figure 1, the higher level's plan implementation output consists of planning inputs to the next tier down.

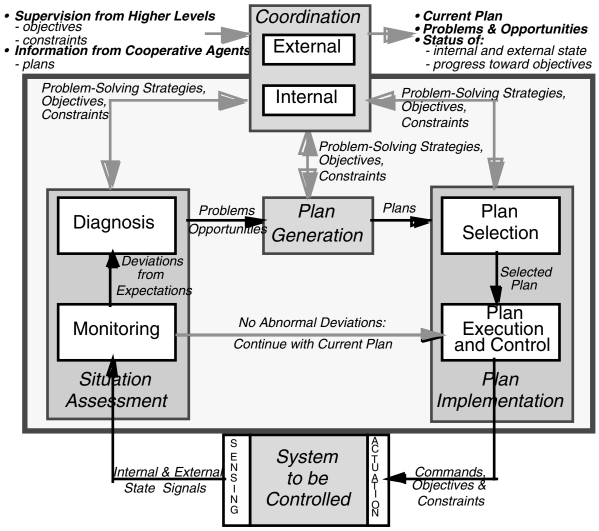

Figure 2: **Functional Decomposition of an Autonomous System**

The *Monitoring Module* validates best estimates of the sensed data and monitors the operation of the system being controlled, as well as the environment in which it is operating, to detect departures from expectations. Any departure could represent a developing problem or a new opportunity, and is passed along to the

Diagnosis Module for interpretation. If no departures are observed, the Plan Execution Module continues to execute the current plan.

The *Diagnosis Module* analyzes departures identified by the Monitoring Module to determine their root cause or, alternatively, their impact on the capabilities of the system being controlled. The Diagnostic Module then initiates the actions required to respond to the root causes or changes in capabilities. If root causes cannot be determined, the Diagnostic Module attempts to reconfigure the system being controlled to retain as much of the systems capability as possible. The diagnosis of what has happened is passed to the Plan Generation Module to determine if the current plan needs to be modified to accommodate the changed circumstances.

The *Plan Generation Module* modifies the current plan when circumstances identified by the Diagnosis Module require such modifications. A new plan may be required in response to either problems or opportunities that have been identified by the Diagnosis module. In support of planning by a superior planning level or in a decision support role for a human planner, it may be desirable for the Plan Generation function to create a variety of plans that trade off among different levels of constraint and/or different objectives. Furthermore, a variety of algorithms may be applied in generating plans and the choice of algorithm is guided by the strategy input. For example, a quick heuristic may be required if a new plan is needed immediately to accommodate a serious (potentially mission or safety critical) problem that has been diagnosed. In other situations, it may be acceptable to continue to pursue the current plan while a more considered search of the plan space is executed in an attempt to refine the current solution to include additional opportunities or to accommodate minor degradations in the capabilities of the system-to-be-controlled.

The *Plan Selection Module* evaluates the utility of the revised plan relative to the current plan (and any other previously generated plans) to determine which plan to execute. A variety of plans may be generated, each based on one of multiple objectives, and given a strategy for selecting among a set of plans, Plan Selection makes the choice of a single plan to be executed. A change in plan may not be warranted if there is no plan in the set of generated plans whose value sufficiently exceeds the value of the current plan. The interpretation of "sufficiently exceeds" is made in the context of strategy inputs from the coordination function. The selected plan is passed to the Plan Execution and Control Module for execution.

Given the current state of the system, the *Plan Execution and Control Module* interprets the current (or selected) plan and issues commands to a subordinate planning level to guide its planning or, for the lowest level of planning, to the physical system-to-be-controlled. Execution of these "set-point" commands results in the pursuit of the current plan. Note that even when there are no detected abnormal deviations, under normal operations there typically will be minor deviations from the expected state. Thus, in addition to providing set-point commands for the pursuit of the current plan, an auxiliary role of the Plan Execution and Control function is to determine plan "perturbation" control commands that will attempt to correct for the range of normally expected deviations of the actual system state from the planned state.

The *External Coordination Module* provides an interface between the system being controlled and other control or controlled elements, which include human operators or users as well as other systems with which the system being controlled interacts. It allows for the receipt of new instructions, which can take the form of new objectives to be pursued and/or new constraints to be imposed on the objectives being pursued. It also provides for reporting out the current plan that is being pursued, the progress that has been made in the execution of that plan, the current state of the system, diagnostic information related to any problems that may have been encountered, and/or any new opportunities that have arisen along the way. Thus, External Coordination is responsible for assembling and transmitting information, deciding: what to transmit, when to transmit it and to whom to transmit.

The *Internal Coordination Module* harmonizes the efforts of the Monitoring, Diagnosis, Plan Generation, Plan Selection and Plan Execution and Control Modules to ensure that

1. Operations proceed smoothly in the absence of any problems or new opportunities,
2. Problems are accurately diagnosed and that the appropriate corrective actions are taken,
3. New opportunities are recognized and appropriately exploited,

4. New plans, when needed, are generated in a timely manner,
5. The transition from one plan to another is effected in a seamless manner, and
6. The current plan is properly executed.

In effecting that harmony, Internal Coordination develops strategies for controlling the other individual modules

1. By monitoring the assessed situation including: progress toward the current solution and the state of both the system-to-be-controlled and the external environment and
2. By taking into consideration any plans developed by other agents and objectives and constraints input from higher level authorities. Included in those strategies are:
   - The criteria for deciding when replanning is required,
   - The time allocated to generating a solution and
   - Cost/objective functions and constraints to be employed in generating a solution.

In each of the functional modules, and each level in the hierarchy, intelligent decisions are required to allocate system resources. Nonetheless, most computational resources will be devoted to the Plan Generation Module. Each level requires a unique mathematical formulation accounting for the specific objective, available resources, and relevant constraints.

Software Architecture

The conceptual implementation of the ADEPT architecture has its roots in the AMMT software, shown in Figure 3. That software was written in C and implemented in a real-time, priority based UNIX environment. The system consists of four tasks:

1. Mission Planner
2. Mission Evaluator
3. Guidance Interface
4. Asynchronous Input Handler

In the context of the current ADEPT architecture, the monitoring and diagnosis functions take place in the Mission Evaluator task, the plan generation function takes place in the Mission Planner task and the plan execution function is in the Guidance Interface task. Since this software has evolved into the current ADEPT architecture, it will be insightful to briefly discuss its implementation.

The Mission Planner task operates at the lowest priority and is responsible for sequencing the activities of a single level in the hierarchy. It considers permutations of the current plan to see if a better plan exists.

The Mission Evaluator incorporates the output from the Planner task into the currently executing plan. It also controls the input to the planner. This input includes which level the planner task should be working with and when it should restart the planning process. Its most important job is to verify the safety and the feasibility of the currently executing plan. If it determined that the environmental conditions have changed so that the current plan is no longer safe, it invokes an immediate replan or provides reactive measures for the near term.

The Guidance Interface operates at the highest priority and is the planner's only interface with the fault tolerant processor. This task converts the “executing” activity into a guidance command that the vehicle's guidance system understands. It also monitors subsystem health, controls subsystems and reads environmental sensors.

The Asynchronous Input Handler is responsible for processing terrain and target messages and updating the map. Moreover, it responds to requests from the sonar and the host ship.

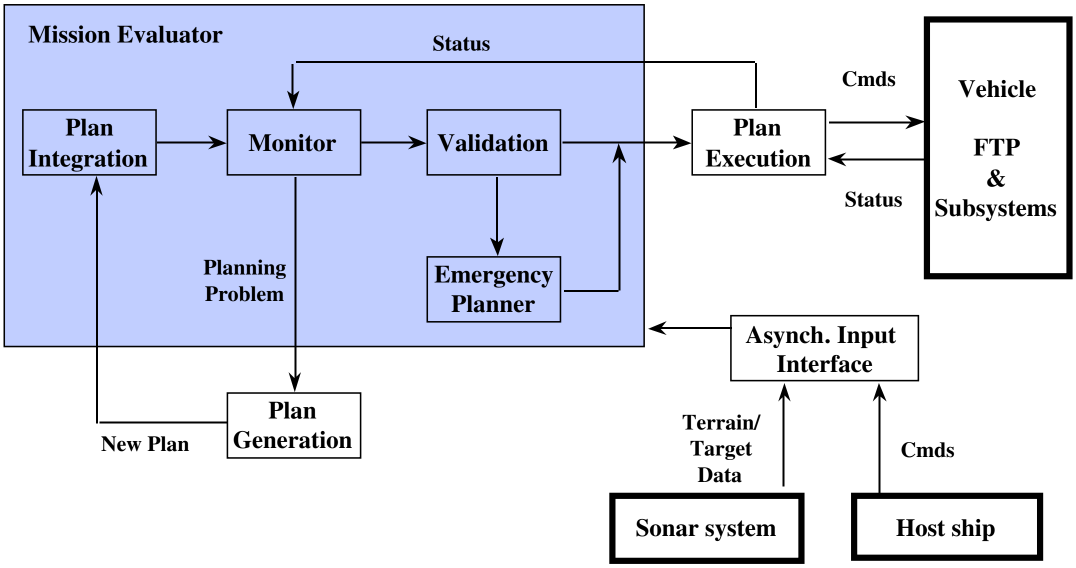

Figure 3: AMMT Software Architecture

The AMMT software has been used as the basis for extending the ADEPT concept into a reusable, object-oriented software framework. In the current architecture, the monitor and diagnosis tasks are separated to allow for a more modular implementation and to allow for the ability to execute the two tasks at different rates if necessary. Conceptually, a single level of the ADEPT architecture consists of an object that has a member function for each of modules that comprises one level of the architecture. Factory and bridge patterns [11] are employed to allow for ease of modification of the functions that are performed in each of the modules. The factory method provides an interface for creating planning objects for different domains while the bridge class separates the implementation from the abstraction. The base planner class itself then just executes the generic functions *monitor()*, *execute()* etc. without regard to the actual implementation being used. The use of the bridge pattern allows the actual *monitor()* function to be activated. Figure 4 shows the layout of the planner class with implementation classes for both an unmanned air vehicle (UAV) and an unmanned undersea vehicle (UUV).

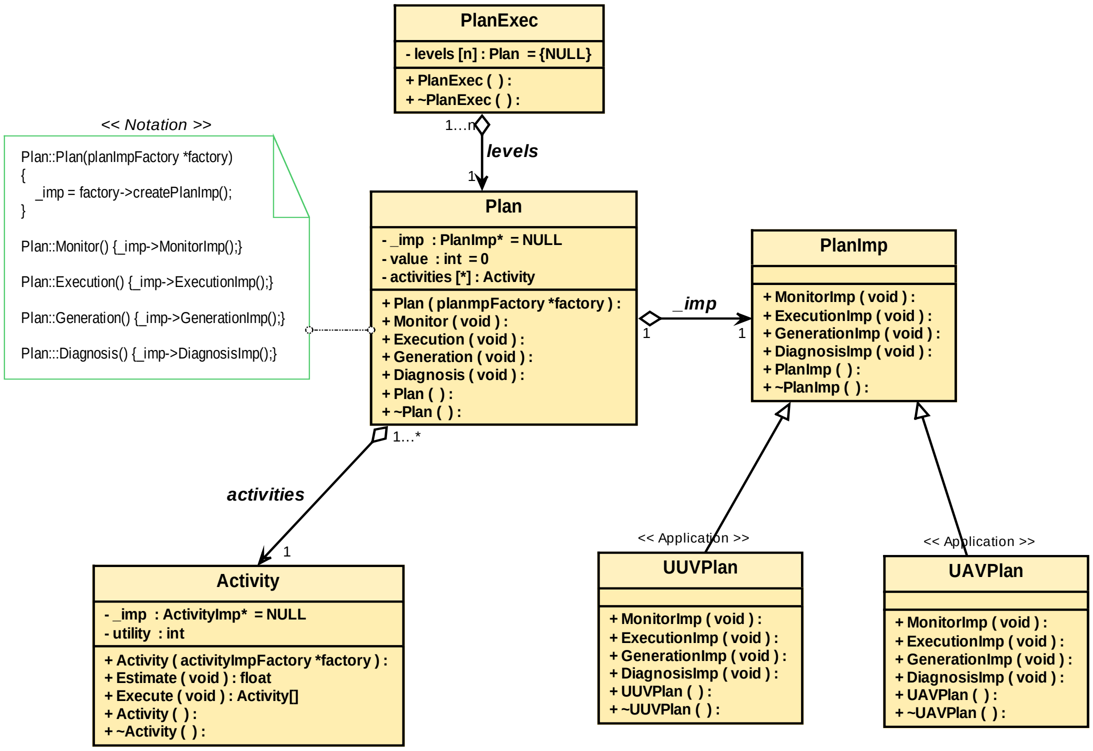

Figure 4: ADEPT Planner class, including 2 domain implementations

The software has been successfully demonstrated in cooperative operations with air vehicles and ground vehicles. This work was supported by Draper internal research and development funding and continues to be extended under externally funded programs. The ADEPT architecture has been embedded into the more encompassing vehicle/ground station architecture developed under internal funding and shown in Figure 5. The ADEPT code runs within the "Mission Planning and Control" block on both the ground station and the autonomous vehicles. While running on the ground station, it is often used to assist an operator in generating plan input objectives. To support this, the planning code can operate in conjunction with a simulated environment on the ground station. When the ADEPT architecture is controlling a vehicle, the communication link to the vehicle passes objectives to the ADEPT software running on-board.

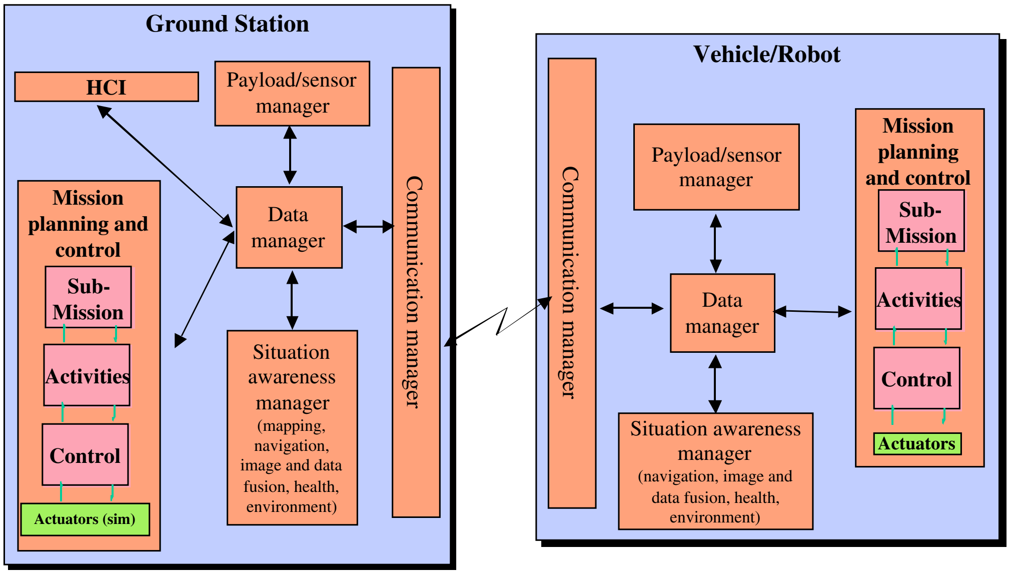

Figure 5: Draper's Autonomous Vehicle Architecture, including ADEPT for mission planning

Applications of ADEPT

The following sections summarize four programs supported by the ADEPT architecture. The first is an ONR program for cooperative multi-autonomous vehicle operations; the second is a NASA program for cooperative constellations of satellites and high altitude UAVs for earth observations. The final two represent higher-level control problems. They include a DARPA program for automating coordinated theater-wide air operations and a program to develop advanced air traffic flow management.

UCAV Intelligent Autonomy

ADEPT was used to provide the mission planning capability for a series of demonstrations performed for ONR, highlighting key technologies necessary to perform ship-based ISRT missions [2, 10]. The planned mission planning capabilities of the system included:

1. dynamic replanning to avoid pop-up threats in a threat-dense environment;
2. planning a two UCAV cooperative geolocation mission, followed by an unmanned ground vehicle (UGV) plan to the geolocated target; and
3. performing a search for a landing platform (surrogate for a ship in sea-state 3) and, working with automatic target recognition (ATR) and target tracking algorithms, maintaining lock on the landing site in order to perform an autonomous shipboard landing.

The first two of these capabilities have been demonstrated.

Supervisory and Autonomous Satellite Operations

ADEPT is being utilized in this NASA program [1] to meet key earth science objectives for rapid response to and tracking of environmental phenomena, and detailed study of the Earth through optimal allocation of resources within a sensor web that includes orbiting satellites and UAVs. Three autonomous mission planning tiers comprise the architecture. The highest, System Level, performs its computations at a centralized control center, and is responsible for allocating the ground based resources. The middle level allocates data collection tasks among individual sensor web elements to meet prioritized input objectives, using centralized or distributed planning. The lowest, Individual Platform Level, allocates the resources for the individual satellite or UAV.

JFACC (Joint Forces Air Component Command)

The ADEPT hierarchical architecture was employed on this DARPA program to plan and execute missions for five wings of aircraft over a seven-day campaign [7, 9]. The levels of the architecture are illustrated and the planning problem solved at each level is illustrated in Figure 6.

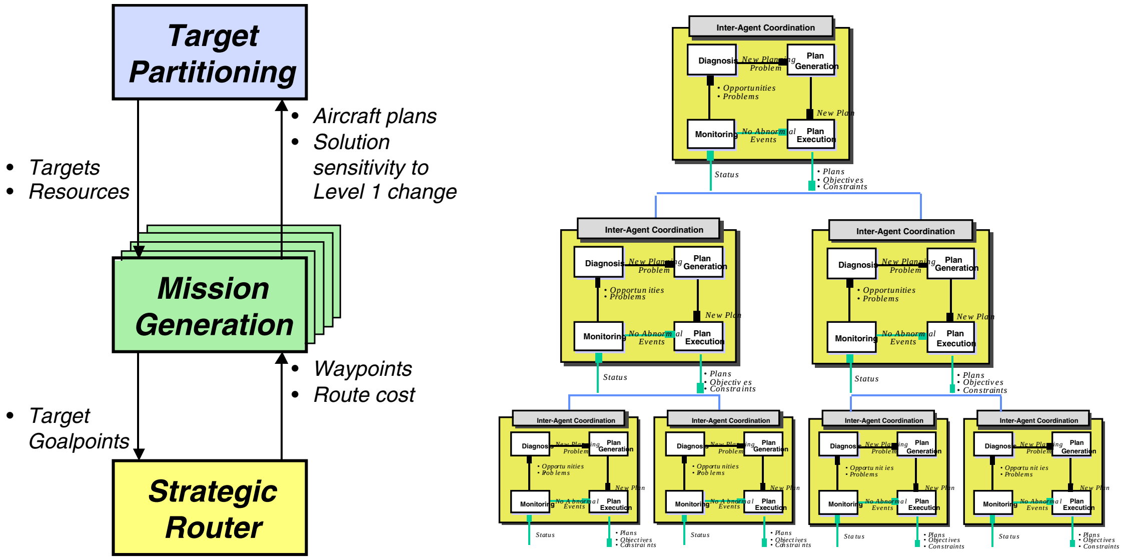

Figure 6: Air Operations Planning and Execution Hierarchy

The *Target Partitioning* level allocates targets and aircraft resources to *Mission Generation* Level. The Mission Generation level forms strike packages over time, assigning specific aircraft to specific targets, employing route costs developed by the *Strategic Router* Level. The Strategic Router Level optimizes paths from bases through tankers to target goalpoints and back, minimizes the route cost comprising a combination of attrition risk and cost of time. The ADEPT architecture allows a truly closed-loop system that is capable of replanning in response to system feedback and new target definitions at intervals as short as a few hours.

Air Traffic Flow Management

During the time ADEPT was evolving, Draper IR&D research on Air Traffic Flow Management (ATFM) was taking place. The ADEPT architecture was used in the development of advanced ATFM concepts and algorithms, some of which are described in [5] and further developed in [13]. ATFM is the top level in the hierarchical decomposition illustrated in Figure 8 and coordinates among planning and control functions at and near the airport and planning and control functions in the en route airspace.

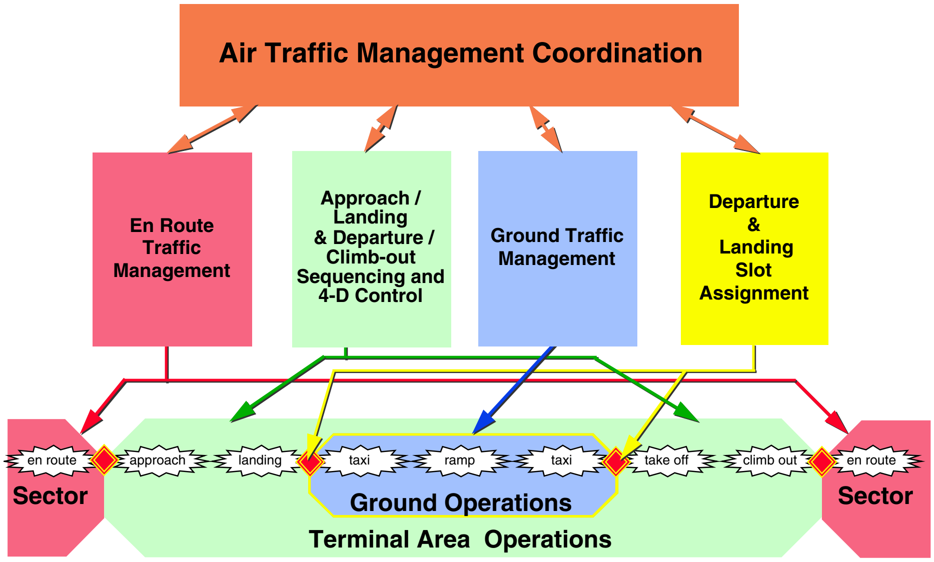

Figure 7: Coordination of Air Traffic Flow Management

Acknowledgments

This material is based upon work supported by the National Aeronautics and Space Administration's Glenn Research Center (contract #NAS3-00163), the Defense Advanced Research Projects Agency (contracts #MDA972-94-C-0011 and F30602-99-C-0206) and the Office of Naval Research (UCAV demo program). The opinions, findings, conclusions and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views or official policies, either expressed or implied, of the National Aeronautics and Space Administration, the Defense Advanced Research Projects Agency, the Office of Naval Research or the U.S. Government.

References

[1] Abramson, M., et al., "The Design and Implementation of Draper's Earth Phenomena Observing System (EPOS)," in *Proceedings of the AIAA Space 2001 Conference*, Albuquerque, NM, August 2001.

[2] Abramson, M.; Cleary, M.; Kolitz, S., "Steps Toward Achieving Robust Autonomy," in *Proceedings of the AAAI Spring Symposium*, Stanford, CA, March 2001.

[3] Adams, M. B., "Functional Analysis/Decomposition of Closed-Loop, Real-time Processes," in *AGARD Lecture Series 200: Knowledge-Based Functions in Aerospace Systems*, AGARD Document # AGARD-LS-200, November 1995.

[4] Adams, M.B., and Beaton, R. M., "Planning and Planning Management for Autonomous and Semi-Autonomous Vehicles," *Proceedings of the AGARD Guidance and Control Panel 51st Symposium*, Knowledge Based System Applications for Guidance and Control, Madrid, Spain, Sept. 18-21, 1990, AGARD-CP-474.

[5] Adams, M., Kolitz, S., Milner, J. and Odoni, A. R., "Evolutionary Concepts for Air Traffic Flow Management", ATC Quarterly, Vol.4. No. 4, 1997

[6] Adams , M. B., Deutsch, O. L., and Harrison, J.V., "A Hierarchical Planner for Intelligent Systems," *Proceedings of SPIE - The International Society for Optical Engineering*, Vol 548, Arlington, VA, April 9-11, 1985.

[7] Adams, M, et al., "Closed-Loop Operation of Large-Scale Enterprises: Application of a Decomposition Approach," in *Advances in Enterprise Control*, Minneapolis, MN, July 2000.

[8] Albus, J. S., Lumia, R., and McCain, H., "Hierarchical Control of Intelligent Machines Applied to Space Station Telerobots," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 24, No. 5, September 1988.

[9] Barth, C.D., "Composite Mission Variable Formulation for Real-Time Mission Planning," SM Thesis, Massachusetts Institute of Technology , 2001

[10] Cleary, M et al., "Hierarchical Decomposition Of Autonomy Requirements For Naval UCAVs," Draper report CSDL-2000-044, July 2000.

[11] Gamma, E., Helm, R., Johnson, R. and Vlissides, J., *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, Reading, MA, 1995.

[12] Hall, W., "Resource-Coordinated Hierarchical Planning for Real-Time Autonomous Systems," SM Thesis, Massachusetts Institute of Technology, 1992.

[13] Hall, W., "Efficient Capacity Allocation in a Collaborative Air Transportation System," PhD Thesis, Massachusetts Institute of Technology, 1999.

[14] Osborne, W. LTC, United States Army, et al. Information Operations: A New War-Fighting Capability, August 1996. (http://www.au.af.mil/au/2025/-volume3/chap02/vol3ch02.pdf).

[15] Ricard, M., “Mission Planning for an Autonomous Undersea Vehicle: Design and Results,” *Proceedings of the Technology and the Mine Problem Symposium*, Monterey, CA, November 1996.