Developing New Technologies for the ARPA-Rome Planning Initiative

Jonathan Stillman and Piero Bonissone, GE Corporate Research & Development Center

A KEY REQUIREMENT OF THE ARPA/Rome Planning Initiative is that the research and technology development must address critical aspects of the overall ARPI vision. The envisioned planning system requires strong capabilities for course-of-action generation, plan refinement, plan analysis, and to a lesser extent, plan briefing. These capabilities in turn require strong technological support in several areas:

- *Course-of-action development* involves building and evaluating a set of possible responses to an emerging crisis. This process is supported by generative and case-based planning, temporal reasoning, and plan visualization.
- *Plan refinement* intertwines initial plan evaluation with plan expansion. It is supported by temporal reasoning, algorithmic and constraint-based scheduling, decision-theoretic analysis, rule-based reasoning, and plan visualization.
- *Plan analysis* evaluates and compares plans along different dimensions, such as operational feasibility, transportation, logistics, or command and control. It is supported by simulation, decision theoretic analysis, case-based reasoning, and plan visualization
- *Plan briefing* allows the user to initiate the planning process, and then explains and justifies the generated, refined, and analyzed plans. It is supported by plan visualization.

THIS SURVEY HIGHLIGHTS SOME OF THE MANY ARPA R&D PROGRAMS THAT ARE FOCUSING ON TECHNOLOGY ISSUES IN PLAN GENERATION, SCHEDULING AND TEMPORAL REASONING, PLAN ANALYSIS, AND THE OVERALL SUPPORTING INFRASTRUCTURE

This article briefly describes some of the ARPI R&D programs that are focusing on technology issues in plan generation, scheduling and temporal reasoning, plan analysis, and the overall supporting infrastructure. Although we focus on individual research efforts, ARPI has encouraged researchers to integrate their work with others to provide integrated capabilities. (See the applications sidebar and related articles in this issue.)

Plan generation

Technology for plan generation is at the heart of the requirements for realizing the ARPI vision.

We anticipate that deployed planning systems in complex environments will rely to varying degrees on several plan-generation techniques: *generative planning* to provide a new, novel solution for those parts of the plan for which existing cases provide no insight; *case-based planning* to build a new plan by adapting and combining previous plans or plan fragments; and a distributed group of human planning experts with diverse, sometimes competing goals, to guide the entire process.

Plan generation is crucial to the ARPI vision, and the initiative's research has yielded both theoretical and practical advances in the technology. A few examples are described below.

Cypress. SRI International has developed Cypress,[1] a planning system designed to cope with the inherent complexity and requirements

0885-9000/95/$4.00 © 1995 IEEE

of real-world domains, including uncertain information, real-time response, intelligent application of standard operating procedures, integration of multiple plans, dynamic plan modification, and interaction with a human planner. Most input to Cypress comes from military databases and includes threat assessments, terrain analysis, apportioned forces, transport capabilities, planning goals, key assumptions, and operational constraints. Cypress includes state-of-the-art subsystems for planning (SIPE-2), reactive execution (PRS-CL), and evidential reasoning (Gister-CL). For example, in one prototype, SIPE-2 generated employment plans and expanded deployment plans to get combat forces, supporting forces, and their equipment and supplies to their destinations on time.

During planning, Gister-CL was used to select military forces based on uncertain information about both friendly and enemy units. The plan was then translated for execution by PRS-CL, which responded to unexpected events with standard operating procedures. When the plan became compromised, PRS-CL invoked SIPE-2 to dynamically modify the plan while continuing execution.

O-Plan. The O-Plan system developed at the University of Edinburgh is a command, planning, and control architecture with an open modular structure that allows experimentation on or replacement of various components.[2] The researchers are trying to determine which functions are generally required in different applications and across different command, planning, scheduling, and control systems. The project's aim is to show how a planner, using extensive domain knowledge and situated in a task-assignment and task-execution (command and control) environment, can allow for better flexible, distributed, collaborative, and mixed-initiative planning. The researchers hope to verify this total systems approach by studying a simplified three-level model with separable task assignment, generation, and execution levels. O-Plan has been applied to logistics tasks that require flexibility of response for changing situations.

Dipart. The Distributed, Interactive Planner's Assistant for Real-Time Transportation Planning (Dipart) allows human planners to simulate and assess alternative strategies for distributed, real-time planning, including those derived from operating systems research.[3] Dipart consists of a network of planning nodes and a simulator. Each node helps a human planner perform some subset of the overall mission objectives; the node forms plans to satisfy the user's goals, coordinates communication with other planners, and alerts the user to reports from agents in the (simulated) world. The nodes also issue commands to the agents in the simulated world, which are then executed. Dipart is thus a very small-scale simulation of an actual crisis-action situation.

Trains. The Trains project at the University of Rochester[4] has focused on the process of interactive human-machine plan construction. To have effective collaborative plan generation, a system must be able to perform many other tasks than plan construction. In particular, the system must be able to identify the implicit information behind a human's suggestions (plan recognition) and must effectively communicate information about plans. To support such generalized plan reasoning, the representation of actions and plans must extend beyond the traditional models used in the planning literature. The effort at Rochester has focused on several crucial plan representation problems. These include the use of explicit temporally-based representation of action and events, allowing external events and simultaneous actions, a representation of plans that captures the key aspects of the plan that are important for communication (such as justification of steps and identification of key assumptions), and an implemented architecture that combines plan construction, recognition and communication to enable mixed-initiative collaborative plan specification.

CAFS. The CAFS system, being developed at GE's Research & Development Center, applies case-based planning techniques to guide force selection for military missions.[5] In one prototype, CAFS receives probes from SRI's Socap planning system. These probes consist of information about a military task, its location, and the expected threat. CAFS uses these probes to retrieve a set of possible

Applications of program synthesis technology

The Airlift Operations Center at Pacific Air Force Headquarters schedules a fleet of 26 C-130 cargo aircraft over the Pacific region. Kestrel Institute applied its ARPI-supported program-synthesis technology to develop the center's first computer-based tools for in-theater scheduling. After building a formal model of the scheduling domain and specifying a scheduler in terms of constraints and problem features, Kestrel personnel synthesized a scheduler in about two hours using the Kestrel Interactive Development System (KIDS, discussed in the body of the article).

Researchers from BBN packaged the Kestrel schedulers into the In-Theater Airlift Scheduler (ITAS), a system with a graphical user interface and a standard database package. ITAS runs on an Apple Powerbook laptop computer, making it useful for operations in the field and in a command center. Users supply ITAS with a list of movement requirements and available assets, such as aircraft, air crews, ground crews, and ports. From this information ITAS generates a schedule, taking into account a variety of constraints on each class of asset. For example, if a port has only two ground crews for unloading, then ITAS will carefully regulate the flow into the port so as not to overload those crews. ITAS displays the generated schedule in a Gantt chart-like form and allows the user to modify various aspects of the situation and schedule. When the user is satisfied that a "flyable" schedule has been generated, then ITAS produces a set of "air tasking orders."

Another application of Kestrel's synthesis technology is underway in a joint effort with the Electric Power Research Institute, Kaman Sciences, and Rome Laboratory. The project focuses on the scheduling of maintenance activities during an outage period at nuclear power plants. Kestrel's program-synthesis technology is being used to model the problem and to generate high-performance schedulers for maintenance activities. Current schedulers used by the utility industry are slow and handle only a small subset of the problem's important features. Safety constraints are extremely important, as well as the efficiency of the schedule, since an outage period can costs millions of dollars per day.

— Douglas R. Smith, Kestrel Institute

matching cases from its case library, and then develops a set of force suggestions based on those cases. Where there is an exact match, CAFS tries to find an available force of the same type used in the retrieved case. If such a force is unavailable, CAFS looks for a similar available force that could also complete the task. Where there is no exact match, CAFS first tries to adapt the retrieved case using the differences between the probe and the retrieved case. Then, starting with the solution from the retrieved case, CAFS tries to find a suitable available force. Ultimately, CAFS returns to Socap the available forces best suited to the task. Once Socap completes the plan, new force selection cases can be extracted from the plan and added to the case library for future use.

Caper. Researchers at the University of Maryland at College Park and the Mitre Corporation are working together to extend the retrieval mechanisms of Caper,[6] a case-based planning system developed at the University of Maryland, to support the Format system being developed by Mitre. Caper uses high-performance-computing techniques to retrieve plans quickly from a large memory that is not preindexed. This approach makes it relatively inexpensive to access memory frequently, and allows memory to be probed flexibly during case retrieval. Caper can issue a variety of queries that result in the retrieval of one or more plans (or parts of plans). These plans can then be merged to solve the problem at hand, and harmful interactions among them can be resolved using annotations on a plan that capture interdependencies among its actions.

Progress and results. Generative planning has been an active area of AI research for decades, but only recently have such systems been able to tackle the large, complex problems found in military planning. For instance, the SIPE-2 subsystem uses about 100 plan operators to describe military operations that can achieve specific employment or deployment goals. The input has around 250 classes and 500 objects, 15–20 properties per object, and around 2200 predicate instances, and the final plans contain about 200 primitive actions.

Significant progress has also been made toward integrating planning and execution, thereby allowing effective replanning. For example, Cypress supports asynchronous runtime replanning with a general-purpose generative planner: when a problem is detected during execution, the system can replan just the affected parts of the plan, while the rest continues to execute. The O-Plan system also provides an architecture that allows the plan-generation and -execution levels to interact.

Progress is also being made toward mixed-initiative planning systems. SIPE-2, for instance, uses a graphical knowledge editor to generate employment plans and expand deployment plans. The Trains system goes furthest in providing such an interface.

Several of these systems have been demonstrated in a wide range of applications. For example, SIPE-2 and O-Plan have been used to plan the actions of a mobile robot, plan the movement of aircraft on a carrier deck, plan the response to oil spills, schedule the production of products, schedule construction tasks, perform project management, perform mission sequencing and control of space probes, and generate tasking for air objectives in air campaign planning.

TEMPORAL REASONING TOOLS LET PLANNERS CAPTURE AND REFINE TEMPORAL DEPENDENCIES IN THE EARLY PHASES OF PLAN DEVELOPMENT.

On a more theoretical level, the Dipart project has improved our understanding of certain strategies for controlling planning in dynamic and distributed settings. ARPI research has also produced plan formalisms that remove some of the inadequacies of traditional state-change-based models. For instance, Ferguson and Allen have developed a plan representation, based on argument structures, that is structurally similar to traditional plan formalisms but allows actions and states to be described in interval temporal logic.[7] Ginsberg has developed a model of approximate planning as a succession of better and better approximations, rather than the construction of provable correct plans.[8]

Temporal reasoning and scheduling

Temporal reasoning and scheduling are critical to the planning process. Temporal reasoning tools let planners capture and refine temporal dependencies in the early phases of plan development. The abstract representation of temporal relationships between canonical elements also helps users build plans that can be effectively scheduled from their inception. Once a plan is sufficiently detailed, scheduling tools then become vital to assessing the plan's feasibility, building it, and maintaining it through the execution phase. This section discusses two ARPI research projects in temporal reasoning (Tachyon and TMM) and two projects in scheduling (Ditops and KIDS/KTS).

Tachyon. Tachyon is a general-purpose constraint-based system for temporal reasoning. Developed by GE's R&D Center, Tachyon provides a powerful and flexible model of events and inter-event constraints. It is capable of reasoning about both qualitative and quantitative aspects of time, and provides support for reasoning over both *and* systems of constraints. The latter are considered to be inherently complex, but arise often in practical planning and scheduling domains where contention for limited resources is important. Tachyon provides both optimal and heuristic solution techniques for such systems. Tachyon can be used as an embedded temporal reasoning system, as a stand-alone file-driven system, or via a graphical user interface.

Tachyon has been applied in ARPI to the refinement and feasibility tracking of the temporal aspects of courses of action (projecting critical steps in a deployment plan to ensure proper interaction between them), and to preserving temporal consistency during force-package structuring and deployment. Also, in a recent ARPI Technology Integration Experiment, Tachyon was integrated with CAFE, GE's case-based reasoning system for constraint-directed force expansion (a companion article in this issue explains the Technology Integration Experiments). Tachyon has also been applied elsewhere to plan recognition tasks, to scheduling for plastics and power systems manufacturing, and to retrieval and situation refinement in a spatio-temporal data management system.

TMM. Honeywell's implementation of the Time Map Manager improves its robustness, user interface, and documentation.[10] The new TMM also gives planners and schedulers extended functionality, including metric and ordering constraints between any two time points, causal reasoning (using the "persistence assumption," projection, and overlap chaining), database monitors (temporal conditions and protections), and optimizations for large temporal databases.

One of TMM's first ARPI applications was a Doctrinal Constraint Checker, a system that tested and (if possible) applied a set of "doctrinal constraints" to a high-level plan for force movements generated by Socap. Constraints that could not be applied were reported to the user, who could relax or ignore the constraint or modify other constraints in an attempt to enforce this one. TMM has also been the basis for planning and scheduling systems in manufacturing, spacecraft operations, distributed image analysis and archiving, and avionics processor and communications scheduling. The avionics application was a particular success: TMM was used to solve a critical scheduling problem for Boeing's 777 aircraft, a multi-billion-dollar project. This application was large, complex, and hard to solve: it contained almost 20,000 activities, several activity types, periodic behavior, and assorted types of temporal constraints.

Ditops. Carnegie Mellon University developed Ditops as a tool for generating, analyzing, and revising crisis-action logistics schedules. Ditops is a mixed-initiative decision-support tool, similar in spirit to spreadsheet programs: the user manipulates sets of scheduling decisions and solution constraints at levels consistent with user-task models. At each step, Ditops applies appropriate scheduling procedures to impose the changes specified by the user, and provides localized consequences of each change. Look-ahead analysis and heuristic scheduling techniques further help the user identify principal causes of observed solution deficiencies, analyze options, and assess a solution's sensitivity to various executional circumstances. The system can repair a schedule in response to unexpected events while minimizing disruption to other parts of the schedule. The mixed-initiative spreadsheet model of operation also makes Ditops well suited to "what if" experimentation. Ditops can relax constraints in several dimensions, allowing it to be used efficiently when scheduling is tightly coupled with planning.

Ditops was implemented using object-oriented representation and programming, which makes it easier to customize the system for different scheduling domains. In particular, Ditops has been easy to integrate with ARPI planning systems. In a Technology Integration Experiment with SRI, Ditops was integrated with the Socap planner to perform resource capacity analysis for higher level course-of-action planning. Some Ditops functionality was also embedded in BBN's Target planning system to perform feasibility checking and to diagnose plan conflicts (Target was part of ARPI's third Integrated Feasibility Experiment; see the article on page 27).

ONE OF ARPI'S GOALS IS TO DEVELOP TECHNIQUES TO HELP PLANNERS COMPARE ALTERNATIVE COURSES OF ACTION FOR SUCH PROPERTIES AS ROBUSTNESS, ECONOMY, AND RISK.

KIDS/KTS. Kestrel's ARPI project explores the transformational development of high-performance transportation schedulers. The researchers used a prototype program-transformation system called the Kestrel Interactive Development System (KIDS) to derive a variety of scheduling algorithms for several military scheduling problems. The transformational approach involves several steps: develop a domain theory, specify a planning/scheduling problem, apply a design tactic, apply optimizations, apply data type refinements, and compile.

After developing a domain theory of military scheduling and various problem specifications, KIDS was used to derive and optimize a variety of global search scheduling algorithms. The resulting code, generically called the Kestrel Transportation Scheduler (KTS), has been run on force-deployment lists generated by planners at the US Transportation Command and other sites. With one such list, KTS scheduled 15,460 individual air movement requirements in 71 CPU seconds; the schedule used relatively few resources and satisfied all specified constraints. (Current scheduling systems would require more than 36 hours to solve a typical deployment list of this size.) KTS performed this well for two reasons: First, the derived pruning and constraint propagation tests are surprisingly strong, which reduces the size of the runtime search tree; although these tests are derived as necessary conditions on feasibility, they are so strong as to be virtually sufficient conditions. Second, the system's specialized representation of the problem constraints, and its specialized and highly optimized constraint operations, allow KTS to explore the runtime search tree at a rate of several hundred thousand nodes per second, almost all of which are quickly eliminated.

Plan analysis

One of ARPI's goals is to develop techniques to help planners compare alternative courses of action for such properties as robustness, economy, and risk. The decision-support tools developed toward this end as part of ARPI include the Course-of-Action Selection Tool (Coast), the Expect COA Evaluator, and a set of tools based on belief nets and influence diagrams.

Coast. Coast, developed at NRaD, is designed to help the process of evaluating and selecting courses of action (COAs) by structuring and documenting the process.[11] The user first selects a set of criteria for evaluating the COAs, such as least risk, lowest cost, or force availability; users can select these from a library, or enter their own. The user then assesses the relative importance of the selected criteria, after which the system performs pairwise comparisons on the highest ranked items to ensure consistency. The user then evaluates the possibility of success for each COA with respect to the chosen criteria; users can also document the rationale behind their rankings via textual attachments, which can then be used to brief the recommended COA. Once all the success ratings have been entered, Coast uses fuzzy logic techniques to combine each rating with the importance of each criterion. The result is a normalized overall evaluation of each COA with respect to each criterion.

Earlier tools for evaluating COAs used weighted-sum techniques to combine importance and success evaluations. While this had the advantage of simplicity, the scores were not normalized, so it was difficult to locate a COA's strong and weak points. Coast's normalized scores make it easier to compare the relative strong and weak points of a plan across categories.

Expect COA Evaluator. Researchers at USC/Information Sciences Institute have developed a COA evaluator using Expect, a knowledge-based-system framework.[12] The Expect COA Evaluator partly automates the development of staff estimates for alternative COAs. It takes as input a high-level description of a COA that specifies the force modules involved, the actions they will perform, and when and where the actions will occur. It uses this information to evaluate heuristically a particular COA from a logistics perspective. This evaluation estimates several factors that are not part of the COA itself, such as the number of logistics personnel required to perform the operation and how long it will take to complete the operation. The current prototype has a graphical user interface with a COA summary window, an evaluation window (that presents the logistics evaluation), and a map that can be clicked on for detailed information.

It is difficult to anticipate all the situations in which a tool like the Expert COA Evaluator might be used, so the framework provides a knowledge-acquisition tool that enables users to modify the system in a timely way: Expect automatically derives a knowledge-based system from abstract domain knowledge and problem-solving methods. During the derivation process, abstract problem-solving methods are specialized by using domain knowledge, and links between domain knowledge and problem-solving knowledge are recorded. These links are then used to form the expectations that guide knowledge acquisition.

Decision-theoretic tools. Rockwell International's Palo Alto Lab has focused on the use of decision-theoretic tools to generate, analyze, and execute plans. DT techniques help planners:

- cope with uncertainty about true states of the world through use of *belief nets* and *influence diagrams* to express risk and uncertainty using subjective probability assessments gathered from planners and decision makers,
- express and reason about the multiple objectives and perspectives that shape strategies using *utility/preference modeling*;
- develop techniques for identifying the most relevant or essential factors in a plan: what uncertainties contribute most to the fragility of their plans. This knowledge will help them focus their efforts to reduce risk and uncertainty. Rockwell has developed new, powerful algorithms for probing large models to compute sensitivities.

Rockwell has developed several large-scale decision support systems for military transportation planning. One of these models, called NEO-COA (Course of Action Analysis for Noncombatant Evacuation Operations), was built in cooperation with ISX, BBN, and the US Pacific Command Crisis Action Planning team. NEO-COA was developed using Demos (Decision Modeling System), a commercially available, Macintosh-based software development environment. It draws on real-world expertise to aid in military operations/evacuation planning for a hypothetical crisis situation.

NEO-COA allows a user to instantiate a generic plan with specific parameter values for locations, forces, and destinations of troops and civilians. It supports scoring alternative plans using different evaluation metrics, such as time to complete the operation or minimizing civilian attrition.

The Demos environment has proven to be effective for developing large decision models such as NEO-COA. It includes several Monte Carlo-based algorithms for propagating uncertainty, and a simple graphical user interface for building Hierarchical Influence Diagrams. HIDs allow users to navigate models at varying levels of abstraction and detail, and to specify and view variables multidimensionally. Such facilities prove especially effective for iterative and collaborative model-building, and for efficient sensitivity analysis needed to enable decision makers to evaluate the impacts of various risks on available courses of action.

Building uncertainty, utility/preference, and sensitivity analysis into a methodology for planning yields several payoffs. Explicit models of uncertainty provide planners with a sense of the risks, opportunities, and ranges of outcomes they face. Decision theory provides an axiomatic basis for choosing courses of action that perform provably well in the face of uncertainty. Including preferences/utilities in models lends them greater descriptive content by adequately describing the concerns and trade-offs of decision makers. The recommendations of the model thereby provide a more convincing basis for action. Finally, sensitivity analysis allows planners to focus attention and to navigate through massive state spaces by identifying the variables and factors that matter most to their decisions.

Supporting infrastructure

As ARPI-developed technology moves toward fielded systems, rigorous evaluation is critical. ARPI has sponsored several activities to help in the evaluation of both individual systems and integrated prototypes.

A working group for "evaluation and critical experiments," with help from other ARPI participants, developed a *Handbook of Evaluation* as a guide to devising evaluation studies.[13] The handbook is a catalog of empirical case studies, using examples from ARPI research where possible. Because ARPI projects range from theoretical work to practical system building, the case studies reflect diverse experimental goals, including exploration, hypothesis testing, sensitivity, comparison, and demonstration. Researchers might choose one or more of these experimental paradigms, depending on their research goals and the kind of evaluative questions they want to ask. Collectively, these case studies offer a framework for evaluating the integration of multiple ARPI projects, whether they combine practical system components or integrate theoretical results with practical systems.

ARPI has also funded the development of various evaluation tools. For example, the Common Prototyping Environment includes tools for instrumenting Common Lisp programs, running experiments, and analyzing the results statistically. It also includes empirical analysis tools specifically designed for the evaluation of AI planning systems. (The article on page 17 offers more details on the Common Prototyping Environment.)

ARPI has also encouraged ad hoc groups of researchers from related projects to develop specific evaluation criteria to further mutual research goals. For example, a committee of researchers recently developed a set of features to evaluate transportation logistics in civilian evacuation plans. These criteria will be used by an automated planner and a knowledge-based plan-evaluation system to guide the generation of multiple, qualitatively distinct alternative plans.

Finally, Rome Laboratory has been evaluating ARPI technologies as they are developed. For example, the first such evaluation compared SIPE-2 and O-Plan2 on a customized set of test problems and on a classic AI planning problem: the "missionaries and cannibals" puzzle. The second exercise was intended to uncover the strengths and weaknesses of each planner.

Uniform access to heterogeneous data sources. Effective planning often depends on identifying and accessing relevant information in distributed heterogeneous data stores. Even after information sources have been identified, there can still be enormous effort and cost involved in dealing with different organizations, access and query languages, and data formats. Furthermore, the data in different sources may use different encoding standards and may require different keys to identify the same people, objects, or locations. Collecting all the necessary information often requires a large amount of time and effort that otherwise would be spent on more productive analysis tasks. Often, the difficulties involved prevent helpful information from being found in time to be useful.

Integrated AI/database technologies will play a significant role in overcoming these problems. ARPI research efforts in this area include Services and Information Management for Decision Systems (SIMS), LIM, and the Cooperative Database (CoBase).

SIMS. USC/ISI's SIMS is an information mediator that provides an interface between users (or application programs) and the distributed, heterogeneous information sources to which they need access.[14] SIMS accepts queries in a uniform language (Loom) that is independent of the distribution of information over sources, the various query languages, the location of information sources, and so on. SIMS determines which data sources to use, how to obtain the desired information, how and where to temporarily store and manipulate data, and how to maintain an acceptable level of efficiency in performing its task.

SIMS applies and extends a variety of AI modeling, planning, and learning techniques and systems to build an intelligent and dynamic interface to information sources, in contrast to the traditional approach of building custom systems. For example, SIMS uses *information source models* that describe both the contents of the information sources and the relationship between those models and the domain model. Also, the system collects information about the contents of information sources as it executes queries against them; this information is abstracted as semantic rules, which are later used to improve the efficiency of the plans generated for new user queries.

SIMS can currently handle Oracle and Mumps databases and Loom knowledge bases. In principle, the SIMS approach supports access to databases, knowledge bases, or even certain application programs.

ARPI HAS SPONSORED SEVERAL ACTIVITIES TO HELP IN THE EVALUATION OF BOTH INDIVIDUAL SYSTEMS AND INTEGRATED PROTOTYPES.

LIM. Many current planning systems use large, heavily encoded records; the plan for even a moderate-sized operation is enormous, containing up to hundreds of thousands of records, effectively precluding any purely memory-resident storage scheme. One of these, at the US Transportation Command, was recently extended to use a relational database — and object-oriented databases in some cases— in place of flat files for some applications. However, the designers of the relational database schema had to adhere fairly closely to the original data format for the plan records, since the planners are quite familiar with the data's current structure. Paramax's LIM project[15] addresses such dilemmas by providing a flexible interface between a knowledge-representation system and a heterogeneous collection of legacy databases that lack a unified semantic schema.

LIM uses the Loom knowledge-representation language to represent both the semantic schema for a relational database and application domain models. LIM augments Loom's terminological component with database mapping constructs, and extends Loom's assertional component with information retrieved from databases. Instances retrieved from databases can be operated upon by both Loom's assertional language and its rule language.

CoBase. The CoBase project at the University of California at Los Angeles is building a knowledge-based extension to standard relational query languages (such as SQL) to provide intelligent access to heterogeneous knowledge and data sources.[16] This includes approximate answers when exact answers are not available, summary answers grouped by category, and answers to high-level conceptual queries. CoBase uses a *type abstraction hierarchy* to provide a structured and scalable approach to query modification, and three types of *explicit cooperative operators* (context-free, context-sensitive, and control) to support query relaxation. This provides multilevel object representation, which is not adequately captured by conventional semantic network and object-oriented database approaches.

The policy for relaxation control depends on many factors, including user profile, query context, control operators, relaxation level, and the number of answers, as well as rules to combine these factors over multiple attributes. A *relaxation manager* combines these rules via certain policies (minimizing search time or nearness, for example) to restrict the search for approximate answers. When an approximate answer is returned, the user may want an explanation of how the answer was derived, or may be presented with an annotated relaxation path. A context-based semantic *nearness* will be provided, allowing the system to rank the approximate answers (in order of nearness) against the specified query.

CoBase can also provide additional information relevant to, though not explicitly stated in, a user's query. It does so using a case-based approach to build a case library of associative links and their terminations among the attributes of previous queries. When a new query is processed, similar query cases and their associative links are retrieved from the case library and modified to provide additional and relevant information to the new query. CoBase has been applied to ARPI and to Medical Image Database, which provides approximate answers to medical queries.

An integrated information access infrastructure. At a 1993 workshop, a Technology Integration Experiment (TIE) between CoBase, SIMS, and LIM was carried out as a knowledge server for the transportation planner, capable of handling imprecise and distributed ARPI transportation queries. In a 1994 workshop, another TIE demonstrated CoBase, SIMS, and LIM in a network environment. In this TIE, CoBase provided approximate and conceptual query answering, SIMS planned and coordinated access to the databases, and LIM served as the distributed knowledge source.

ARPI RESEARCHERS HAVE MADE significant progress toward developing the necessary planning and scheduling technologies for the initiative's visionary system. Crisis action planning still presents a challenging set of problems, however, and further success depends on continued progress, including both theoretical results and technological advances.

For example, mixed-initiative systems require better techniques for plan justification, plan explanation, plan reuse, contingency and adversarial planning, managing planning's dynamic nature, managing inconsistent or incomplete information, and developing and implementing detailed models of mixed-initiative interaction.

Also, both plan development and reactive planning and scheduling could also benefit from improved techniques for real-time sensitivity/robustness analysis.

Finally, new techniques for managing the huge amounts of data generated during planning and execution will be critical to the next generation of systems, even though some related areas — such as data fusion and data access — may be beyond the initiative's scope.

Acknowledgments

We have tried to correctly describe the work of many individual researchers. Any mistakes are unintentional. Several people made significant contributions to this article; James Allen, Bill Swartout, and David Hart were especially helpful.

Further reading

The proceedings of the 1994 ARPI workshop (published by Morgan Kaufmann) covers many of the research endeavors not discussed here, including plan reuse to avoid repeated plan failures, rational resource allocation during plan revision, a new backtracking technique (a relaxation of dynamic backtracking) to search among possible planning solutions, advances in mixed-initiative planning, case-based schedule evaluation, time-critical scheduling, techniques for learning planning knowledge, case-based system development from existing archives, case-based plan adaptation, and approximate query answering.

References

1. D. Wilkins et al., "Planning and Reacting in Uncertain and Dynamic Environments," to appear in *J. Experimental and Theoretical AI*, Vol. 7, No. 1, 1995.
2. A. Tate, B. Drabble, and R.B. Kirby, "O-Plan2: An Open Architecture for Command, Planning and Control," in *Intelligent Scheduling*, M. Fox and M. Zweben, eds., Morgan Kaufmann, San Francisco, 1994.
3. M. Pollack et al., "The Dipart Project: A Status Report," *Proc. 1994 ARPA-Rome Laboratory Knowledge-Based Planning and Scheduling Initiative Workshop*, Morgan Kaufmann, San Francisco, 1994.
4. J. Allen et al., "The Trains Project: A Case Study in Building a Conversational Planning Agent," to appear in *J. Experimental and Theoretical AI*, Vol. 7, 1995.
5. P. Bonissone and S. Ayub, "Representing Cases and Rules in Plausible Reasoning Systems", *Proc. 1994 ARPA-Rome Laboratory Knowledge-Based Planning and Scheduling Initiative Workshop*, Morgan Kaufmann, San Francisco, 1994, pp. 305–316.
6. J. Hendler et al., "Massively Parallel Support for Case-Based Planning," *Proc. 1994 ARPA-Rome Laboratory Knowledge-Based Planning and Scheduling Initiative Workshop*, Morgan Kaufmann, San Francisco, 1994, pp. 317–324.
7. G. Ferguson and J. Allen, "Arguing About Plans: Plan Representation and Reasoning for Mixed-Initiative Planning," *Proc. AI Planning Symposium*, 1994, pp. 43–48.
8. M. Ginsberg, "Approximate Planning," *Proc. AI Planning Symposium*, 1994, pp. 68–73.
9. J. Stillman, "Dual-Use Applications of Tachyon: From Force-Structure Modeling to Manufacturing Scheduling," *Proc. 4th Ann. IEEE Dual-Use Technologies & Applications Conf.*, 1994.
10. M. Boddy, J. Carciofini, and G. Hadden, "Scheduling with Partial Orders and a Causal Model," *Proc. Space Applications and Research Workshop*, 1992.
11. R.W. Larsen and J.S. Herman, "Course-Of-Action Selection Tool: Coast," *Proc. 1994 ARPA-Rome Laboratory Knowledge-Based Planning and Scheduling Initiative Workshop*, Morgan Kaufmann, San Francisco, 1994.
12. C. Paris and Y. Gil, "Expect: Intelligent Support for Knowledge-Based System Development," in *Knowledge Acquisition for Knowledge-Based Systems*, N. Aussenac et al., eds., Springer Verlag, New York, 1993.
13. P. Cohen et al., "Handbook of Evaluation for the ARPA-Rome Laboratory Planning Initiativ," *Proc. 1994 ARPA-Rome Laboratory Knowledge-Based Planning and Scheduling Initiative Workshop*, Morgan Kaufmann, San Francisco, 1994, pp. 519–536.
14. Y. Arens et al., "Retrieving and Integrating Data from Multiple Information Sources," *Int'l J. Intelligent and Cooperative Information Systems*, Vol. 2, No. 2, 1993, pp. 127–158.
15. J. Pastor and D. McKay, "View-Concepts: Persistent Storage for Planning and Scheduling," *Proc. 1994 ARPA-Rome Laboratory Knowledge-Based Planning and Scheduling Initiative Workshop*, Morgan Kaufmann, San Francisco, 1994, pp. 375–384.
16. W. Chu and H. Yang, "CoBase: A Cooperative Query Answering Facility for Database Systems," *Proc. 1994 ARPA-Rome Laboratory Knowledge-Based Planning and Scheduling Initiative Workshop*, Morgan Kaufmann, San Francisco, 1994, pp. 351–362.

Jonathan Stillman is a computer scientist at the Information Technology Laboratory at the General Electric Corporate Research and Development Center. His research has centered on automated reasoning and algorithmic issues arising in planning and scheduling, temporal reasoning, and spatio-temporal data management. He received a BA in music from the State University of New York at Oneonta and his MS and PhD degrees in computer science from the State University of New York at Albany. He is a member of the ACM. He can be reached at GE Research & Development Center, Bldg. K1, Room 5C31A, Schenectady, NY 12301; stillman@crd.ge.com.

Piero Bonissone is a computer scientist at the General Electric Corporate Research and Development Center. His research interests include artificial intelligence, expert systems, rule- and case-based reasoning, simulation, fuzzy logic, and fuzzy controllers. He is the editor-in-chief of the *International Journal of Approximate Reasoning*, associate editor of the *International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems*, and one of the editors-in-chief of the *Handbook of Fuzzy Computing*. He received the BSEE from the University of Mexico City, the MS in EECS, MS in ME, and PhD in EECS from the University of California at Berkeley. He is a member of the IEEE, AAAI, ACM, and NAFIPS. He can be reached at GE Research & Development Center, Bldg. K1, Room 5C32A, Schenectady, NY 12301; bonissone@crd.ge.com.