Planning with Templates

Doug Dyer, *Active Computing*

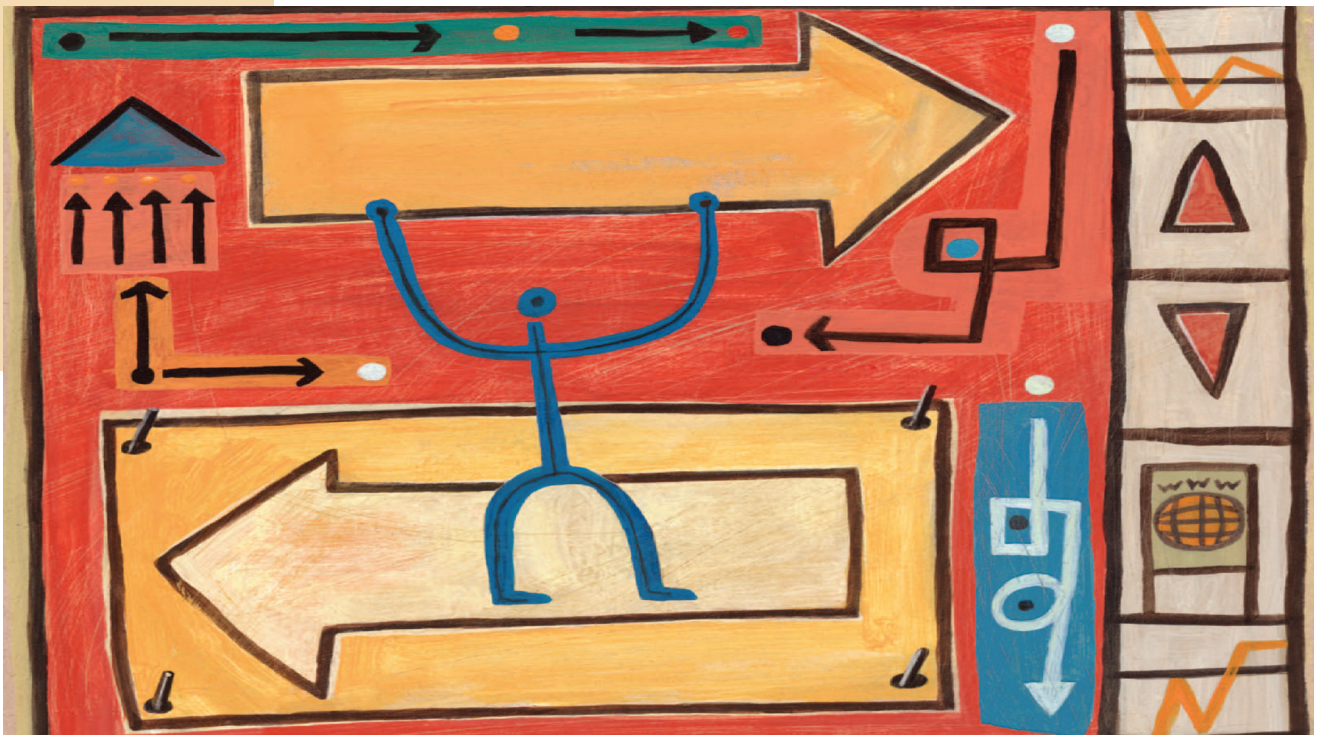
Steve Cross, *Georgia Institute of Technology*

Craig A. Knoblock, *University of Southern California*

Steven Minton, *Fetch Technologies*

Austin Tate, *Artificial Intelligence Applications Institute*

When we talk about planning with "templates," we mean standard operating procedures we've learned or been trained to use for solving typical problems and as a starting point for solving novel problems. These structures contain variables relevant to an activity and current variable values that affect a problem or have been chosen in a problem-solving instance. With templates we can configure domain-independent



1541-1672/05/$20.00 © 2005 IEEE
Published by the IEEE Computer Society

planning algorithms, making them applicable to many different problem domains. By making the template explicit for the user in the form of a GUI, we can facilitate mixed-initiative, user-centric systems that help maintain awareness of complex and dynamic situations, share information across the network, and solve problems incrementally and iteratively.

Template benefits

Templates have three key advantages. First, they reduce complexity. Templates let us exploit expertise and use default activities rather than reason every time from first principles. Because they include all the relevant variables, templates make clear the requirements for solving a problem even when solution methods aren't known. In addition, when we encounter novel problems, templates let us differentiate the familiar parts from the new parts. This makes it possible to have the machine handle routine details while humans focus on the novel parts.

Second, they reduce uncertainty. As standard operating procedures, templates provide some enumeration of the probable actions of agents that, for whatever reason, are out of communication and can't be observed. For distributed, coordinated planning and execution control under these circumstances, knowing what collaborating agents will likely do in a given situation is a significant advantage.

Finally, they facilitate process improvement. Successful templates can be learned and reused for problem solving and for other purposes such as learning better strategies, training, and selecting more appropriate activities in real situations.

Mixed-initiative systems

Fully autonomous planning is a worthy goal, but the complexity of many real-world problems favors mixed-initiative systems, which combine human judgment with the machine's ability to remember, serve information, calculate, and handle the details. Despite great advances and continuing research, machines usually have less knowledge and poorer situational context than humans. So, fully autonomous planners are applicable to a more limited set of problems than mixed-initiative systems. Furthermore, successful mixed-initiative systems' behavior is easier to understand and more readily adopted by end users. For both practical and technical reasons, mixed-initiative systems are the logical bridge to greater autonomy, given our starting point: today, almost all problem solving is manual.

Mixed-initiative systems imply a requirement for a kind of human-machine dialogue, and successful systems also provide decision-making information for incremental problem solving. The simplest form of dialogue is a GUI, and the simplest GUI is a form listing variables and values. During problem solving, these interfaces present information from external sources and record choices the user makes, ideally showing implications of choices in addition. Because the problems are complex, they must be decomposed so that choices are simple enough to make. Choices made constrain future choices, leading to an iterative search that eventually discovers a solution. During problem solving, successful systems alert their users to state changes that affect the problem, including other players' decisions.

User-centric development

The software development process's importance is often underestimated. The traditional approach is to contract the entire project with a software developer, who then must become a domain expert. User-centric approaches can improve this inefficient process. Users, who generally have much more domain knowledge than a developer, can specify and group key variables and describe functions and interfaces that are meaningful to them. We deem important any technology that lets users contribute their knowledge and even build part of the system for themselves.

Vertical, knowledge-based software applications are often extremely important for organizations. However, their limited markets aren't attractive to major software manufacturers, and traditional development makes these systems prohibitively expensive. To make progress, users will have to provide part of the solution so that developers can focus their efforts on programming rather than domain expertise. Adopting user-centric development also gives users more control and facilitates incremental development with incremental payoff, making software projects easier to manage. We're beginning to see user-centric applications with tailorable user interfaces such as smart forms.

Active Templates

Beginning in 1998, DARPA sponsored the Active Templates R&D program. Active Templates aimed to create prototypes integrating situational awareness, experience and standard operating procedures, and planning to achieve goals amid uncertainty and constant dynamics while keeping all players coordinated. The program chose to apply the technology to military mission planning and execution control, particularly in the special-operations domain.

The Active Templates research agenda was guided by real problems addressed by real people who had real reactions when the prototypes didn't meet their expectations. DARPA also required that the project advance the state of the art, not just engineer point solutions. The domain experts and researchers who participated in the program experienced many failures and reworked their systems constantly to move the technology forward. You can measure the program's success by technology transition: nearly every prototype has been picked up for further development, and some are operationally used.

In this issue

The four articles in this issue are exemplars of the advances that came out of the Active Templates program and are relevant to related technology development efforts.

In "Conditional Constraint Networks for Interleaved Planning and Information Gathering," José Luis Ambite and his colleagues describe a system for creating planning templates that exploit a hierarchical, conditional constraint network. Heracles II incorporates a constraint propagation algorithm that supports incremental problem solving without imposing any order on user decisions. The system templates present key information to users at the point of decision making, reducing human effort through automated discovery and information management. As the user makes planning choices, Heracles II uses the choices' implications to update only the affected information, greatly improving scalability. Constraints are propagated asynchronously, because of either a user decision or dynamic external information, and cycles are handled appropriately. Heracles II has been applied to travel planning and geospatial-data integration.

In "Comirem: An Intelligent Form for Resource Management," Stephen Smith, David Hildum, and David Crimm describe a user-centric, mixed-initiative scheduling and resource allocation system that meets real-world needs of both planners and those executing a plan. Astute observation of human planners has resulted in key insights:

- Users want to solve problems at different abstraction levels and different degrees of automation for different circumstances.
- Tailored graphical interfaces are effective

for supporting mixed-initiative dialogue.
- Executers can't easily tolerate huge plan changes, so localized plan changes, applied incrementally, are more appropriate than reoptimization.

Comirem epitomizes excellence in mixed-initiative scheduling systems: the machine handles details such as feasibility checking, resource tracking, conflict identification, and, if the user prefers, conflict resolution. Graphics communicate different decisions' impact. The user is free to plan manually, specify key constraints, select from a set of suggested solutions, or let the machine select a solution. Comirem is a full-featured tool that supports military logistics in the air-shipping domain.

In "Applications of SHOP and SHOP2," Dana Nau and his colleagues describe SHOP2, an efficient hierarchical-task-network (HTN) planner based on ordered task decomposition that allows interleaving of subtasks. This decomposition lets the planning focus on the tasks that will execute first, an important idea for integrating planning and execution in dynamic domains where state changes might invalidate a plan. Standard operating procedures relevant to a particular domain are used to configure the planner. SHOP2 and its predecessor, SHOP, have been used to create planning systems for augmenting an evacuation planner, evaluating terrorist threats, controlling unmanned aerial vehicles, and other applications. Nau and his colleagues have graciously made SHOP and SHOP2 available as open-source software. SHOP-based research continues in automated learning, compiled domains, information-dynamic planning, and planning under uncertainty.

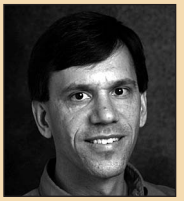
In "Authoring Templates with Tracker," Alice Mulvehill describes a forms-based application that takes user centricity to the next level by enabling ordinary users to participate in application development. Like a spreadsheet, Tracker is a tool for both building and using applications. During development, the developers tightly link variables to form elements and use a dialog box to specify properties, constraints, and formulas. This results in a hierarchically oriented form display made persistent with an XML file. During use, forms store information in XML files, facilitating information sharing. Throughout use, many insights occur as users discover the actual process, and they can refine the domain ontology and the application. New applications can incorporate templated parts of previous applications. All these features combine to facilitate rapid forms-based application development useful for planning, coordination, and workflow. Tracker has been used to develop an automated clearance tool and several other applications.

DARPA and other research organizations continue to see potential in intelligent systems for military command and control and many other areas. Spin-off projects outside DARPA include Advanced Concept Technology Demonstrations (www.acq.osd.mil/actd/index.htm) and an ongoing series of joint experiments, and system development by the US Department of Defense. Despite current prototypes' success, much work remains, particularly to reduce tailored systems' development cost. With more experience in developing, deploying, and using mixed-initiative, user-centric systems, we can accelerate the evolution of our machines from networked, calculating terminals to intelligent planning assistants able to handle details and even major problems. That's a vision worth pursuing. ■

The Authors

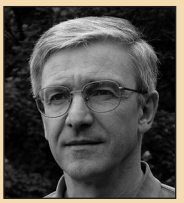
Doug Dyer is a computer engineer and the president of Active Computing, a small company supporting the US Departments of Defense, State, and Homeland Security. His research interests include decision aids, automated planning and coordination technology, and intelligent simulation. He's a retired Air Force officer and former program manager for the DARPA Active Templates program. He received his PhD in computer engineering from the Air Force Institute of Technology. Contact him at Active Computing Inc., 12712 Nathan Ln., Herndon VA 20170; doug.dyer@activecomputing.org; www.activecomputing.org.

Steve Cross is a vice president of the Georgia Institute of Technology and the director of the Georgia Tech Research Institute, the nonprofit, applied-research arm of Georgia Tech. His research interests include industrial and systems engineering, knowledge-based planning and scheduling, and software engineering. He was previously the director and CEO of the Software Engineering Institute and a DARPA office deputy director. He received his PhD from the University of Illinois at Urbana-Champaign. Contact him at the Georgia Tech Research Inst., Georgia Inst. of Technology, Atlanta, GA 30332; steve.cross@gtri.gatech.edu; www.gtri.gatech.edu.



Craig A. Knoblock is a senior project leader at the University of Southern California's Information Sciences Institute and a research associate professor in computer science. He's also the chief scientist for Fetch Technologies, a company commercializing some work developed at USC. His research interests include information agents, information integration, automated planning, machine learning, and constraint reasoning. He received his PhD in computer science from Carnegie Mellon University. Contact him at the USC Information Sciences Inst., 4676 Admiralty Way, Marina del Rey, CA 90292; knoblock@isi.edu; www.isi.edu/~knoblock.

Steven Minton is the chief technology officer of Fetch Technologies. His research interests include machine learning, planning, and constraint satisfaction. He received his PhD in computer science from Carnegie Mellon University. He founded the *Journal of Artificial Intelligence Research* and served as its first executive editor. He has also served as an editor of *Machine Learning*. He's a fellow of the AAAI. Contact him at Fetch Technologies, 2041 Rosecrans Ave., Ste. 245, El Segundo, CA 90245; minton@fetch.com; www.fetch.com.



Austin Tate is the technical director of the Artificial Intelligence Applications Institute and holds the Personal Chair of Knowledge-Based Systems at the University of Edinburgh. His research area includes knowledge systems and workflow process standards activities, and DARPA supports his O-Plan and I-X planning research. He's a fellow of the Royal Society of Edinburgh. Contact him at the Artificial Intelligence Applications Institute, Univ. of Edinburgh, Appleton Tower, Crichton St., Edinburgh EH8 9LE, UK; a.tate@ed.ac.uk.