C. PROJECT DESCRIPTION

C.1 Relevant results from past NSF support

CCR-0100040, Theory Revision and Related Problems in Learning Theory, $213,265, 8/2001–7/2004, Goldsmith, PI (current). The work supported by this grant falls into two categories: computational learning theory, which is not directly relevant to the current proposal, and the learning of utilities, specifically the early work on POET, the programmable online elicitation tool. It has also partially supported our initial work on the Semistructured Probabilistic Database Management System (SPDBMS). Both POET and the SPDBMS will be refined and used in the work presented in this proposal.

The computational learning theory work has resulted in several papers [GSST02a, GSST02b, DGS03], as has the work on learning utilities [RHGD02] and the database work [DG02, GZD03, DGZ03]. Three other papers are in preparation by invitation to special issues of journals. The grant has supported three graduate students and one undergraduate RA; three other undergraduates and one high school student have also been involved in the research.

CCR-9610348, The Complexity of Markov Decision Process Problems, $50,000, 6/97 – 5/99, Goldsmith, PI (completed). This work produced four refereed journal papers [MGLA00, LGM01, LGM98, GOR00] and five papers in major theory and AI conferences (including one invited paper) [LLS+99, MGA97, GLM97, GM98, GOR98]. All these papers are relevant to the proposed work and concerned planning with Markov decision processes. The grant supported one graduate student, plus travel for three others. The high complexity of planning with Bayesian net models and even of approximating optimal plans in this setting [LGM01] is one of the major motivations for the work proposed here.

IIS-0097278, Truszczyński, PI, Marek and Finkel, co-PIs (current). We are studying theoretical and practical aspects of answer-set programming formalisms (building on research originated under previous grant). We obtained new results on stable-model and well-founded semantics of logic programs [MR01, LT01, LT03, LT02, MR03, MNT03]. We have developed a new ASP formalism, logic of propositional schemata [ET01b, ET01a]. We are investigating local-search methods in ASP [LT03]. We also initiated research on Constraint Lingo, a high-level constraint-modeling language for which ASP serves as processing back-end [FMT02].

IRI-9619233, Truszczynski, PI, Marek, co-PI (completed). We investigated uses of default logic as a knowledge-representation formalism and as a computational tool. Our implementations DeReS and TheoryBase (a benchmarking suite) [CMMT99] had significant impact on the area.

Our study of default logic led us to a new computational paradigm called *answer-set programming* (ASP) [MT99]. ASP departs from the single-intended-model approach that is dominant in the logic programming area, adopting a semantics in which a program (theory) is assigned several intended models (answers), each representing a different answer (or solution). We have proposed a specific ASP formalism, *DATALOG with constraints* (**DC**) [ET00]. Since its introduction, ASP has attracted wide interest in the knowledge representation and logic programming communities (one of co-PIs co-organized in Sept. 2002 a very well-attended Dagstuhl seminar on the topic). We also studied semantics for default logic and related nonmonotonic formalisms [DMT98, DMT99, DMT00b, DMT00a]. Our approach is based on algebraic approach proposed for logic programming by Fitting and Przymusinski. We studied the complexity of nonmonotonic reasoning [MNR97, MNR96, CT99, MNR99, Tru99]. Finally, we studied a nonmonotonic formalism for representing integrity constraints in databases [MPT99b, MPT99a, MPT02].

Both grants (IRI-9619233 and IIS-0097278) supported several graduate students. Two (East and Pivkina) have received PhD degrees; one obtained an MS degree (Han). We currently support two MS students and one PhD student. We are working with a few others and will support two additional students starting with Summer 2003. In addition, we have worked with several undergraduate students.

C.2 Present proposal — Introduction

We propose research on **planning under uncertainty**. Specifically, we will study frameworks, methods, and tools for modeling and solving complex realistic decision-theoretic planning problems. Such problems are ubiquitous; we will focus the practical component of our research on problems appearing in the areas of education and social work. This proposal is distinctive in that it (1) closely integrates core qualitative (logic-based) and quantitative (probabilistic) techniques developed in AI for the tasks of knowledge representation and reasoning, (2) develops efficient and uniform database support for the modeling and planning tasks, and (3) crosses disciplines, involving both computer scientists and experts in the areas of education and social policy. In each of these respects, the project is unique. Our overarching objectives in the project are to: 0

1. Advance the theory and practice of decision-theoretic planning by extending standard formalisms of *partially observable Markov decision processes* (**POMDP**s) and *Bayesian networks* (**BN**s) with logic-based knowledge-representation modules to model and process constraints on states, actions, preferences, utilities and policies; by developing planning algorithms generating optimal and sub-optimal policies, taking advantage of all types of qualitative information by which the POMDP/BN model is populated; and by providing systematic database support specialized for the tasks of collecting, storing and manipulating all the major model components including states, actions, probabilistic data, utilities, preferences and constraints.
2. Demonstrate the feasibility of extended decision-theoretic planning models in societally important areas of education and social welfare by applying them to (a) academic advising in higher education (an application guiding our research and development work in the area of decision-theoretic planning), and (b) the complex and volatile case management of Welfare to Work participants (our primary application to show effectiveness of the proposed approach). We will develop software to assist people engaged in complex planning that must be responsive both to personal needs and preferences and to specific regulatory, economic and educational environments.
3. Enhance cross-disciplinary collaborations between researchers and students in the areas of computer science, educational policy studies, and instructional systems design and evaluation by jointly developing and applying tools to support the advising work of case managers in the Welfare to Work program.

Specific research directions we will pursue in support of these broad objectives are: (1) **Modeling:** developing an expressive formalism for modeling application domains along with advisee preferences and utilities; (2) **Planning:** developing planning algorithms based on data stored in such models; (3) **Database support:** developing uniform database support for collecting, storing and manipulating probabilistic data and constraints; (4) **Applications:** demonstrating the feasibility and effectiveness of our approach in the areas of academic advising and Welfare to Work. These research directions are detailed in Section C.4.

This research has significant intellectual merit. It addresses fundamental questions in IT concerning modeling complex, real-world planning domains, and proposes new techniques to address them. Our approach, to combine quantitative and qualitative techniques in AI, has several novel elements and strong potential to better handle the large amounts of data necessary to populate models of practical application domains and to support planning algorithms that will scale up.

This research will have broad impact in several ways. (1) It represents a novel collaboration between Computer Science, Educational Policy Studies and Instruction and Curriculum Studies. (2) It will train students from these fields in skills necessary for cross-disciplinary research work. (3) It will advance social goals, particularly with regard to the Welfare to Work program, enabling more efficient delivery of services, more effective use of services, and a higher success rate. We provide more details in Section C.5.

Our project team is well positioned to accomplish the goals of this project. The team includes five faculty members of the Department of Computer Science at the University of Kentucky, with expertise covering the broad spectrum of areas in logic-based AI, quantitative AI, databases, programming languages, and software-system development. The team also includes two faculty members from the College of Education, working on problems related to the Welfare to Work program, a key application area for this project. Dr. Goldstein (Department of Educational Policy Studies) and Dr. Mazur (Department of Instruction and Curriculum) will provide domain expertise necessary to model the Welfare to Work application and study the effectiveness of our IT methods in the social work domain. The project will also include several graduate and undergraduate students from the departments of Computer Science and Educational Policy Studies. We give more details in Section C.6.

Our management plan includes resources provided by both the Department of Computer Science and the College of Education, with carefully delineated responsibilities assigned to each member of our research team. This plan is detailed in Section C.8.

This project is designed for four years, a time frame adequate to accomplish all major goals of the project: (1) establish representation models and planning algorithms based on qualitative and quantitative approaches, (2) develop software prototypes tailored to our application areas and (3) demonstrate the soundness and technical feasibility of our approach. In the next phase (under a separate project to be undertaken after the one we propose here is completed) we plan to deploy software resulting from this work in a practical setting of Welfare to Work and to conduct comprehensive studies of usability and effectiveness of these tools through a series of field studies addressing, among others, questions raised at the end of Section C.4.4.

C.3 Preliminaries

Mathematical models. As a novel contribution, we propose to integrate two existing, complementary, areas: (1) the quantitative approach of Markov decision processes and its variants, partially observable Markov decision processes, and Bayesian networks, and (2) logic-based, or qualitative formalisms of logic programming with negation and versions of first-order logic.

A *Markov decision process* (**MDP**) is a 4-tuple $\langle \mathcal{S}, \mathcal{A}, t, r \rangle$, consisting of (1) a set of states $\mathcal{S}$, (2) a set of stochastic actions $\mathcal{A}$ with effects described by (3) the function $t : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, which determines, for each state s and action a, a probability distribution over possible next states s', and (4) a reward or utility function $r : \mathcal{S} \to \mathcal{R}$. We consider here only MDPs over finite state spaces and finite sets of actions.

A *partially observable MDP* (**POMDP**) is a 6-tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, o, t, r \rangle$, where $\mathcal{O}$ is a set of possible observations, and o is a deterministic or stochastic function from states to observations. Researchers use POMDPs to model situations where not all information is observable. For instance, given only a student's transcript, the student's IQ is not directly observable.

A **policy** for an MDP is a total function from states to actions. A policy for a POMDP might map observations to actions, or it might map both the current observation and the time the process has been running to actions (this form is relevant in a time-limited process like Welfare to Work), or it might map the sequence of observations since the process started to actions.

An **optimal policy** is one that has the optimal expected cumulative reward over a particular length of time. In some cases, future rewards are discounted; discounts form part of the optimality criterion. Unfortunately, finding optimal policies for POMDPs, particularly optimal functions from sequences of observations to actions, is intractable.

In some MDPs and POMDPs, states can be represented as vectors, where each entry in the vector represents the value of a feature, or **fluent**. This representation is convenient when the effect of actions can be factored so that each fluent is affected only by the values of a few others. We represent these MDPs and POMDPs using a *dynamic Bayesian network* (**BN**), which is a directed, bipartite graph and a set of conditional probability tables. The states of the MDP or POMDP correspond to vectors of fluent values at time t and time $t + 1$, represented by two sets of graph nodes. The edges go from nodes in the first set to nodes in the second and represent conditional dependence. The BN conditional probability tables quantize the dependencies described by the graph. These tables may be represented by more succinct data structures.

The job of a **planning algorithm** is to construct a policy, based on the model (the BN, the utility function, and possibly the relevant constraints). A **policy** is a total function from system states, from states $\times$ time, or from sequences of observations, to actions. The **planning problem** for stochastic systems is to find a policy that yields the highest expected utility under a specified metric (such as total expected utility or total discounted expected utility).

To model and reason about constraints, we adopt the computational paradigm of *answer-set programming* (**ASP**) [Nie99, MT99] that has recently emerged from earlier work on nonmonotonic logics such as default logic [Rei80, MT93] and logic programming with negation [Cla78, ABW88, Prz88, GL88]. The idea behind answer-set programming is to represent constraints defining search problems by means of theories in some logic so that models of these theories represent solutions. Consequently, to implement answer-set programming one needs methods to compute models rather than proofs, a significant departure from more standard approaches to knowledge representation.

In the project we will study and incorporate with BN models two basic answer-set programming formalisms: logic programs with negation under stable-model semantics [GL88, MT99] and the logic of propositional schemata [ET01b]. In each case, a **program** is a collection of **clauses**, which can be viewed as non-classical implications (logic programs) or disjunctions (logic of propositional schemata) of atoms and their negations (literals). Both individual clauses and programs can be viewed as **constraints** defining particular classes of models. In the context of logic programming, we consider stable models [GL88]. In the case of the logic of propositional schemata we consider a variant of the standard semantics of Herbrand models [ET01b].

We have chosen these formalisms for this project because they are known to be effective in representing broad classes of search problems that appear in AI [Bar03, ET01b] and because each has fast algorithms for processing programs [ELM$^+$98, SNS02, ET01b]. Significant ongoing research on effective implementations of these algorithms and closely related research on fast algorithms to compute models of propositional theories [SKC94, GN02, MMZ$^+$01] (applicable to computing both stable models of logic programs [LZ02, BL02] and models of propositional-schemata programs [ET01b]) is bound to further enhance answer-set programming formalisms in modeling and processing search problems.

Application areas. The applications we will deal with involve people getting advice from professionals. An **advisee** in some **program** goes to an **advisor** who produces a plan or policy, which — barring any changes in the advisee's utility or the domain — will specify the actions to be taken throughout the program.

The plans must take into account (1) the rules of the program, which we call the **domain**, (2) the advisee's preferences, (3) the advisee's current situation, and (4) predictions of the likelihood of various outcomes of any suggested actions.

We will build our prototype software based on an academic advising application, where the advisors are University faculty, advisees are students, the program is the Bachelor's degree. A high-quality plan must take into account (1) domain rules, such as University requirements, (2) student preferences, such as aversion to a particular professor, (3) the student's current situation, including the transcript of courses taken so far, and (4) predictions of the likely grade resulting from this particular student taking a particular class.

Our major application is more complex. Welfare to Work is a social welfare initiative of the federal government implemented at the state level to move welfare recipients into self-sufficiency within the labor force. In the Welfare to Work program, the advisors are case managers and advisees are participants. A high-quality plan must take into account (1) domain rules, including legal requirements, (2) participant preferences, such as a particular line of work, (3) the participant's current situation, elicited through extensive interviews, and (4) predictions of the likely outcome if a particular participant engages in a particular sponsored activity.

The Welfare to Work program has many interlocking pieces. Through *Temporary Assistance for Needy Families* (**TANF**), federal funding is provided as block grants to individual states, which then develop programs that meet federal and state regulations but are responsive to state-level economic, demographic, educational and employment conditions. These programs are administered in Kentucky by the Cabinet for Families and Children through the *Kentucky Transitional Assistance Program* (**KTAP**) and the overlapping *Kentucky Works Program* (**KWP**) [CFC03, CN99]. KTAP and KWP provide cash assistance to eligible families; basic education, job placement and training to un- and underemployed adults; and support services such as health care, food stamps, housing and transportation subsidies. These programs are implemented at the county level in Kentucky and vary across counties depending on community resources such as social work, health care, childcare, educational and employment services available within each county.

To enter Welfare to Work programs, participants must have their eligibility determined through a series of assessments conducted by case managers. Case managers assess economic need, family structure and dependents, levels of work, basic and educational skills, ability to learn, attitudes toward employment, and personal situation and perception of self, and then determine the goals, actions and supports needed for the participant to become self-sufficient. These actions and supports are contracted between the case manager and participant and are then reviewed and revised on a regular (often monthly) basis for the duration of participation in Welfare to Work. As case managers develop goals and plans with their participants, they must balance federal requirements for employment activities with those activities allowed by the state but not counted in the federal laws (such as basic education and rehabilitation services). Availability of services, programs and employment vary by locale (metropolitan, urban, and rural) and by economic conditions. The decision-making problems for case managers are that (1) the regulatory environment is complex and volatile and (2) local conditions and personal circumstances of the Welfare to Work participants change frequently, necessitating regular revision of goals and action plans. Heavy case loads add to the stress: a range of ten to one hundred cases per month with a mean of 35-40 per case manager in one metropolitan county. Furthermore, individuals have a maximum lifetime eligibility of 60 months in which to receive welfare support, so they often choose to drop in and out of Welfare to Work to preserve months of this safety net for later use. Participation is therefore not necessarily continuous or compliant.

C.4 Proposed research

Our research has four main components: (1) **Modeling,** (2) **Planning,** (3) **Database support,** and (4) **Applications.** The following subsections describe each of these components in detail.

C.4.1 Modeling

In this section we address research issues related to modeling and outline ways in which we plan to approach them. We consider three research tasks here: (A) to propose and study a formalism for modeling application domains, (B) to propose and study formalisms to represent utility functions, and (C) to design and implement software tools to support the process of eliciting domain knowledge and user utilities.

A. Modeling application domains. Although the formalism of POMDPs is an expressive modeling tool and theoretically capable of representing essentially all dynamic domains, it has well-known practical limitations, primarily the prohibitively large amount of data needed to fully instantiate a POMDP model. Consequently, neither the process of data collection nor planning algorithms scale up well.

One way to reduce the number of conditional probabilities needed to represent a POMDP is to represent it by a collection of BNs exploiting the inherent structure of the POMDP to represent states as vectors of fluent values. The multiplicity of BNs arises because each action is represented separately. Alternately, the BNs could be combined into one influence diagram.

We will adopt this BN representation for our project. However, we propose significant extensions to this basic model to further facilitate collecting, representing, and using data in planning. Specifically, we propose to introduce and study the notion that that actions can be decomposed into their basic components, or *elementary* actions, and that actions, states and policies can be specified or refined by means of constraints.

Actions in complex systems may be factored into several smaller actions such as taking two independent courses. Restricting attention to elementary actions and their effects dramatically reduces the amount of data to be collected and represented. However, in order for factoring to be usable, we must develop techniques to combine information about the effects of elementary actions in order to approximate the effects of the corresponding composite action. We will study this question in depth. In particular, we will study ways to represent and approximate dependencies between pairs of actions and use this data to extrapolate more complex dependencies, extending the work of [DGK98, MHK+98, bKEKD02, GVK02].

We propose a promising and novel way to facilitate concise domain representations: to integrate *constraints* into the model of BNs. Constraints can be used to disallow states, actions (which elementary actions can form an action), and policies (which sequences of actions are allowed). To model and represent constraints, we will use the logic-based knowledge representation formalisms that we have been developing and investigating (in grants described above). We will focus on two specific formalisms: logic programming with stable model semantics [GL88] and the logic of propositional schemata, developed by one of co-PIs and his student [ET01b].

In each of these formalisms, programs are viewed as concise representations of their models (stable models, in the case of logic programs, and Herbrand models, in the case of programs in the logic of propositional schemata). To specify the space of states, we construct a logic (propositional-schemata) program that captures all known constraints on states in such a way that models of the program correspond to states. Further, we represent valid actions by building programs that constrain sets of elementary actions that can be used together to form an action. A similar approach can also be applied to constrain the set of policies; one difficulty is that a policy is not just a set of some basic values, but a function on states.

Thus, we propose to represent application domains by a collection of BNs together with another collection of logic and propositional-schemata programs, each specifying a separate set of constraints. We refer to this model as *Bayesian networks with constraints* (**BN+C**).

We propose to investigate the BN+C model, including answering basic questions concerning its expressive power and the complexity of its associated reasoning tasks. Although qualitative constraints can be handled in terms of additional fluents, states, and modified preferences and utilities, that approach leads to increased complexity of the overall representation. We believe that logic-based approaches, which do not require that new concepts be introduced and represented in a model, will prove more effective.

Recent research demonstrates that logic programming and the logic of propositional schemata are capable of capturing broad classes of constraints [SNS02, MT99, ET01b, MR01]. Moreover, each of these formalisms has fast algorithms for computing models of programs (for us, valid states, actions and policies). We will investigate techniques for constructing concise programs customized to planning, representing valid states, actions and policies and their integration into the BN+C model. Successful attempts to combine some forms of quantitative and qualitative information in the formalism of probabilistic logic programs (some by on of the co-PIs) [KS92, NS94, KDR00, DS00, DRS01] indicate that close integration of the two approaches is feasible.

B. Modeling and evaluating utility functions. The goal of decision-theoretic planning is to compute policies that with high likelihood lead to states that are preferred in some sense, or, in a more general setting, have high expected utility. The problem of modeling, eliciting and manipulating preferences and utilities is of major importance for decision-theoretic planning [Bou02]. The main obstacle is that of the large amount of data that is needed. Our proposed solution represents partial (in some sense, *atomic*) information concerning utilities; we will design algorithms that, based on this atomic data, approximate the utility for any given state.

We will pursue two approaches along these lines. First, we will continue our research on the *Programmable Online Elicitation Tool* (**POET**) [RHGD02]. That work introduces a data structure, a *forest of decision trees*, and corresponding algorithms to compute the utility. This work assumes an additive d-separable utility function and severely restricts the types of relationships between preference items that can be expressed easily. We will study both the representation and the elicitation process.

In a parallel effort, we will study the use of logic programs to represent hard constraints and preferences. We will build on our recent work [BNT03] on representing and computing preferences among states (vectors of fluent values), which generalizes an earlier formalism of CP networks [BBHP99]. CP networks reduce the amount of information about preferences among states (through the mechanism similar to that used in BNs) and provide methods to decide which of two states is preferred.

We will further generalize CP networks. First, we will use answer-set programming formalisms such as logic programs with stable-model semantics and the logic of propositional schemata to specify the space of all legitimate (feasible) states. Recent research shows that these formalisms are well suited for expressing broad classes of constraints [SNS02, ET01b, MR01]. Without this extension, CP networks are unable to directly exclude states from consideration. Second, rather than using CP networks, we will represent preferences by means of extended logic program clauses (*preference clauses*) of the form

$$c_1 > \ldots > c_k \leftarrow a_1, \ldots, a_n, \mathbf{not}(b_1), \ldots, \mathbf{not}(b_m).$$

where a_i, b_i are values of fluents A_i and B_i, and c_i are values of some fluent C that depends on A_i and B_i. The clause has the following intuitive reading: Among states that have values a_i for fluents A_i and have value other than b_i for fluents B_i, the most preferred are states with value c_1 for fluent C, then those with value c_2, and so forth. We call programs consisting of preference clauses *preference programs*. Our recent preliminary work showed that this syntax provides much more flexibility in modeling dependencies between fluents than is possible with CP networks, and captures more complex preferences [BNT03].

We will focus on algorithms to compare states and to compute most preferred ones. Furthermore, we will extend this approach to the case when the task at hand is to acquire and manipulate utilities, which are functions on states, and not just preferences. This extension is important because most decision-theoretic planning algorithms require a utility function.

Finally, we will also investigate ways in which formalisms based on preference programs could be combined with representations currently used by POET.

C. Tools for eliciting domain knowledge and utilities. The BN+C model that we propose for representing dynamic domains consists of two interdependent parts: (1) a collection of BNs representing the stochastic evolution of states, and (2) constraints on components of the model and its evolution (states, actions and policies). We need to elicit that information from experts and represent it in a precise, accessible, unified language.

The BN model of an application requires the following information: a set of nodes or fluents, the dependencies between them, the applicable actions, and the conditional probabilities that quantify those dependencies. The description of available actions may be more complex than a simple set. For instance, students may not be allowed to take more than 24 credits per semester, the classes they take may not meet simultaneously, and prerequisites must be fulfilled. Welfare to Work has similar restrictions, as well as time and budgetary limits on an individual's tenure in the program.

The number of experts is quite large (thousands of professors, hundreds of case managers), so comprehensive interviews to elicit the model are infeasible. Further, interviews are not directly generalizable to other applications. We propose to elicit models online using application-independent software. The models we derive must track a "moving target" as domain specifics change (academic advising: curricula and requirements; Welfare to Work: laws and budgets). Eliciting models online can automate this tracking.

We propose to develop elicitation tools that have both graphical and natural-language interfaces. These tools will establish basic vocabulary (naming the relevant nodes for the BN and describing the form of necessary constraints) and then elicit the needed information. In particular, we will build (1) a vocabulary/node and action elicitor, (2) a dependency elicitor, (3) a constraint elicitor, and (4) a probability elicitor.

We have initial results in each of these areas as well as techniques and data structures for storing the related information. For constraints, we have designed a language called Constraint Lingo that allows us to describe row-based and column-based constraints in tabular constraint-satisfaction problems [FMT01, FMT02]. We have built software that converts such descriptions into logical formalisms such as ASP systems (logic programming and logic of propositional schemata), uses a logic solver to solve the constraint-satisfaction problem, and then presents the results as a table. This software also generates human-readable reasoning steps to explain the solution.

For the elicitation of other data, we are currently developing and testing POET, the Programmable Online Elicitation Tool. POET represents an advisee's preferences over a finite set of attributes, each of which has a finite set of values. The utility of an attribute value may depend on that value's conjunction with values of other attributes. We assume that the utility functions are additively separable.

POET represents the utility functions as dependency trees, where each branch represents a conjunction of attribute values and a utility for that conjunction. The value of a *state* (the current description of the advisee's progress) is the sum of all utilities for branches consistent with that state.

As one way to facilitate utility elicitation, we exploit the concept of an *archetype*. Archetypes model canonical types of advisees and can be used to "jump start" a process of specifying utilities. For instance, in the academic advising domain, one archetype is *"I want to go to graduate school"*. POET prompts the advisee to choose the archetype that fits best and then displays a pre-set utility function. The advisee can adjust the set of relevant attributes and values, specify dependency structures, and assign or reassign utility.

A prototype of POET exists. We will further study problems of utility representation and computation, we will continue our development of the concept of the archetype and we will create interfaces between POET and Constraint Lingo to ensure seamless integration of constraints with other types of data.

In order to ensure that the tools we intend to develop are adequate, our the work on elicitation software will involve members in both Computer Science and the College of Education. This collaboration will allow us to address some key human-computer interaction problems. For instance, the psychological and marketing literature raises questions about how people hold irrational beliefs about probabilities [TK74]. Our probability elicitor will present probability distributions in a variety of formats (tables, visuals, verbal descriptions) and will recognize and respond to apparent inconsistencies in its input, recognizing when the expert is getting tired or bored or is disagreeing with herself or with other sources of related information. In addition, it will allow us to develop interfaces to databases of relevant Welfare to Work information that we will utilize in the project; we discuss these in section C.4.4.

C.4.2 Planning Algorithms

Standard approaches to planning with MDPs use dynamic programming or linear programming; POMDP methods include value iteration, policy iteration, and search heuristics [LLS+99, Han98, MKKC99, MPKK99]. Although these approaches could be adapted to the case of BNs, we will not pursue them. First, we expect that techniques specifically tailored to the model of BNs will be more effective. Second, our model, BN+C, combines BNs with constraints; direct translations from existing algorithms for MDPs and POMDPs do not address constraints.

This section presents two research problems: (A) to design and evaluate heuristics for planning with BNs subject to constraints (planning with BN+C model), and (B) to integrate KBMC (a model-reduction technique discussed below) with our constraint representations, implement the resulting algorithms, and study their behavior in constraint-rich real-world situations.

A. Heuristics for planning. Because the problem of finding *optimal* plans for very large systems represented by BNs is intractable, the only realistic approach is to apply heuristics to find *acceptable* plans. The research problem is to design and evaluate heuristics for planning with BNs subject to constraints. We plan to look at five heuristic methods.

(1) **Integrating planning with constraints.** Current work on integrating planning and constraints transforms an entire planning problem into a SAT instance [DL01]. In contrast, we plan to use constraints both for preprocessing (KBMC, discussed below) to limit both the number of fluents considered in the BN and the available actions (inspired by Feng [FH02]), and also to restrict the search required by planning algorithms. This approach allows us to maintain the semantic information inherent in the model and may allow us to reduce problems to a tractable size.

(2) **Hierarchical planning.** Researchers have taken several approaches to hierarchical planning. One is to reduce a larger planning problem to independent, smaller problems [LK02]. Another involves semi-independent subproblems and subsequent negotiations for resources [HMB+98, GVK02]. Our approach will be to to treat individual subgoals as major goals, plan for those, and then integrate the solutions into one commonly acceptable solution. For instance, all students must satisfy a variety of core university requirements (two semesters of foreign language instruction, two of social studies), plus their major requirements. Although a student cannot necessarily satisfy all outstanding requirements in a given semester, we can first determine how to satisfy each individual requirement and then use the utility of each individual solution to combine them into a coherent policy. Similar reasoning applies in the Welfare to Work application.

(3) **Hot starting.** Academic advisors tend to suggest the same solutions to all students within a certain category (say, fifth-semester students who have satisfied all core requirements and a have good GPA); they modify these suggestions slightly for students who diverge from the category. Such pre-planning does not scale well; it tends to put too many students in the same courses. However, some randomness or history tracking can flatten the distribution of students in courses. Similarly, instead of solving each advisee's planning problem from scratch, we can preprocess broad categories of start states (representing categories of advisee histories) and starting archetypes (representing standard advisee utilities) to find good (maybe even optimal, since this computation is offline) plans for each category and then compute advisee-specific modifications. We conjecture that suitable categories and archetypes will let us find near-optimal plans for most advisees with reasonable time and space complexity.

(4) **Plan reuse.** Many advisees might seek advice at the same time. We intend to implement web-based software that can handle many requests simultaneously. One way to achieve this goal is to reuse plans, with necessary modification, for batches of similar advisees. In the absence of utility change, one policy should be sufficient for a advisee's career, but both the advisee's utilities and domain specifics are subject to change. If such changes are relatively small, plans might be updated rather than recalculated from scratch. Updating requires that we store plans; storage might be impractical for an application with thousands of advisees.

(5) **Local search techniques.** Solving the planning problem involves search over a solution space of all possible plans of a specified type for a given BN. We can define the notion of the *neighborhood* of a solution. Given one solution, we can search its neighborhood and substitute a better neighbor if one exists. This strategy may mire the search in a local optimum. The art in local search algorithms involves

defining the neighborhood appropriately, choosing one of the better neighbors, and escaping local optima by jumping out of the neighborhood or restarting from different initial solutions. Such local-search techniques are considerably faster than other planning techniques, but they have no guarantee of near-optimality.

We have had some initial success in finding good finite-memory policies for POMDPs [LLS+99]; others have also used local search techniques [MKKC99, MPKK99, KDM00, HZF02]. We expect that our use of constraints will produce smaller neighborhoods and that hot starts (described above) will locate the neighborhood in a good region of the search space.

B. Integrating knowledge-based model construction. Our second research problem in planning is to reduce the BN to a manageable size. The size of the BN has a dramatic effect on the run time of any planning algorithm. Applications often contain irrelevant actions: an illiterate Welfare to Work participant will not sign up for college-credit courses within six months, nor will a mono-lingual CS major sign up for Russian literature in the original.

Knowledge-based model construction (**KBMC**) is a collection of methods for building situation-specific stochastic models [Bre92]. Typically, KBMC methods are used to help inference algorithms run faster by constructing smaller models, based on available information about the current state. The wide variety of KBMC approaches include probabilistic logic programs, relational BNs, and probabilistic relational models [KDR00]. KBMC has also been applied to planning problems [LM00, PKMT99, Had99, Pfe01].

KBMC methods can be broken into two broad categories: those used to produce smaller models before the planning algorithms start, and those used within planning algorithms to trim the model dynamically. We propose to study both approaches.

To facilitate KBMC, we will collect, store and manage a wide array of situation-specific information about the application domain: BN dependencies, conditional probability tables, and domain constraints will all include *context* describing associated situation-specific information. For instance, given a BN dependency "student performance in the Database course depends on prior performance in Data Structures and Discrete Mathematics courses", we can store different *conditional probability tables* (**CPT**s) for the `Databases` node of the BN depending on the student's major. Similarly, student performance in a course might exhibit different properties based on who is teaching the course.

Given the database of such context-specific information, a KBMC query is a request to construct a specific BN, given the known information about the advisee and about the current state of the domain. For example, a KBMC query may be "build an academic advising BN for the current semester for a Computer Science student." The BN builder would have information about courses, schedule, and faculty. The KBMC algorithm will produce an explicit description of a model that best approximates the situation at hand. KBMC involves two phases: (1) computing the smallest BN fragment that can serve as the domain for any reasonable policy for the particular advisee under the given circumstances, and (2) finding the conditional probabilities to fully populate the network. We will research these issues. In order to compute the BN fragment, we will study heuristics to determine the reachability of nodes in the graph and to generate sets of relevant nodes. To find relevant conditional probabilities, we will propose and study a representation of "relevance data" and formal methods to reason about it.

The second type of KBMC occurs within planning algorithms in the form of a heuristic search that uses available situation-specific information to prune the planning search space by dynamically modifying the stochastic model. Work in that direction has been done recently by Hansen and his group [FH02, ZH02, HZF02] for MDPs. We plan to develop KBMC algorithms for heuristic search when planning with BNs.

C.4.3 Database support

This proposal addresses every aspect of decision-theoretic planning in complex domains with uncertain information, including management of data. Data management is of paramount importance due to the complexity, diversity and amount of data needed to build models of practical domains.

The decision-theoretic framework we will develop includes a variety of different types of data: BN dependency structures, conditional probability tables, user preferences and utility functions, and advisee and domain constraints. All this information needs to be obtained from experts, data, or advisees, and it must be stored and delivered upon request during planning.

Conditional probability tables for BNs form the largest mass of data we need to handle. First, our BN models have a large number of random variables. Second, the conditional dependencies among the variables may be quite complex, leading to large sizes of individual CPTs. Third, our KBMC techniques require that we store many situation- and advisee-specific CPTs for each node in the network.

Our knowledge-elicitation tools and planning algorithms must deal with this enormous mass of data efficiently. We cannot rely on ad-hoc data management methods. Instead, we will delegate all data-management tasks to a database-management system designed specifically to provide support for storing and retrieving diverse probabilistic information. This systematic integration of database techniques is one of novel elements of our project.

We will build on our earlier work where we proposed a flexible database model called *Semistructured Probabilistic Objects* (**SPO**s) for both point [DGH01] and interval [GZD03] probability distributions. An SPO contains information about a probability distribution of one or more discrete random variables. We will use an SPO to represent conditional probability tables, and we will associate descriptive non-stochastic data (the **context**) with each distribution. We have designed a markup language, SPOML, to represent SPOs as XML documents. One of our PhD students has implemented a prototype *Semistructured Probabilistic Database Management System* (**SPDBMS**). It processes XML documents that contain SPOML data, stores SPOs in a *relational database system* (**RDBMS**) with a specially designed schema, and executes queries in SP-Algebra [DGH01].

SPDBMS will be the backbone of our modeling and planning software suite. It will help us achieve the following goals:

- **Consistent representation.** SPOs and their associated SPOML format represent a flexible framework for storing the probability distributions needed for BNs and MDPs. All model-building and planning software will rely on a single data format and use a single API to the SPDBMS.
- **Clear separation of duties.** This approach factors data-management aspects out of KBMC and planning algorithms and delegates them to the SPDBMS.
- **Efficient data management.** The SPDBMS will use traditional database algorithms for optimizing query execution and delivery of data. Because the SPO model and SP-Algebra differ significantly from their relational counterparts, the problems of optimizing data storage and query processing require additional research, described briefly below.

Several issues related to SPDBMS have already been resolved, but we still need to address key problems for SPDBMS to provide high-quality database support for our project.

1. **Convenient and efficient data storage.** The most efficient means of storing XML data currently place the data in a relational database. A variety of general-purpose translations of XML data into relational database form have been proposed [TDCZ02]. Because SPOML is a specific markup language and we have exact knowledge about the structure of SPOML instance documents, we can propose specific translation mechanisms of SPOML into relational database schemas. Our current version of SPDBMS imple-

ments one such (non-optimal) translation. We plan to propose, implement, and test several SPOML-to-relational-DB translations and select the most efficient one(s) for further incorporation into SPDBMS.

2. **Translation SP-Algebra queries.** We represent atomic retrieval and modification queries as operations in SP-Algebra. The SPDBMS must translate each SP-Algebra operation into a sequence of SQL queries to the data, with postprocessing to assemble the resulting relational data into SPOs. The efficiency of the SPDBMS depends on the quality of this translation and postprocessing. We plan to investigate, implement and test a variety of translations and to select the best ones for the SPDBMS.

3. **SP-Algebra query optimization.** Most queries for probabilistic information are represented as compound expressions in SP-Algebra. We can apply traditional query-optimization methods to the problem of finding optimal executions of SP-Algebra queries. Although the RDBMS underlying the SPDBMS implementation can optimize individual SQL queries, query optimization is also valuable as we construct SP-Algebra queries. In particular, relational representations of SPOML data lose information specific to the nature of the data, so the cost models the RDBMS uses to optimize SQL queries fail to take into account statistics that reflect the properties of the SPO collections (as opposed to their relational representations), such as average number of instances per SPO. Relational cost models also ignore the probabilistic nature of the stored information, for example the fact that no SPO has more than one instance with probability greater than 0.5.

   We propose to study the question of optimization of SP-Algebra queries in detail. We are currently investigating SP-Algebra query-rewrite rules. We will propose, implement, and test several models to estimate the computational cost of executing particular query plans. We will then build a query optimizer for SPDBMS.

By solving these problems, we will achieve efficiency at every layer of the SPDBMS: from bottom (relational representation of SPOML data) to middle (translation of SP-Algebra operations into SQL) to top (optimization of SP-Algebra queries).

C.4.4 The Welfare to Work application

In the first stage of the project, the policy research will determine what decision paths case managers use to guide participants in (1) their best plan of action, (2) activation of welfare eligibilities, and (3) strategic use of limited welfare entitlements. We will also investigate how case managers incorporate changes of policy and participants' changing life circumstances and goals in revising individual plans.

As decision-theoretic planning tools are developed, the research will test whether case managers can use those tools in ways that they find (1) supportive of their work efficiency; (2) consonant with their understandings of successful strategies for participants to move from welfare to work; and (3) beneficial for optimizing and revising action plans. We will consider how the advice given by case managers compares to the plans our tools generate. Specifically, we will:

1. Elicit the model from case managers employed in one service region under the supervision of an organization called LexLinc. We have attached a letter of cooperation from LexLinc.

We will discover the decision paths case managers use in the extended advising process for developing goals and plans for participants. This advising process entails a series of revisions over many months as participant circumstances change, actions have outcomes, and the regulatory and economic environment changes. We will elicit data on the advising process from case managers through interviews and focus group methods [Pat01, Pat97, MH94] that walk case managers through their standard KTAP and KWP assessments with simulated participant cases. These case managers advise KTAP and KWP participants through the following agencies supervised by LexLinc: Department for Community Based Service, Community Action Council, and Family Care Center.

2. Transform multi-source, longitudinal databases on Welfare to Work participants into a comprehensive and uniform database. We have received preliminary permission to access two enormous databases: (1) the Kentucky Welfare Reform Administrative Database (KY Cabinet for Families and Children), which has data on about 150,000 participants from 1997 to the present, and (2) the Kentucky Reform Evaluation Panel Study (University of Louisville Urban Studies Institute, Dr. Rod Barber, PI), which has data on about 2,000 participants from 1998 to the present. (We have attached a letter of collaboration from Dr. Barber.) We will use this information to populate our model.

3. Test our software that automates planning with Welfare to Work case managers to determine whether our tools improve planning-time efficiency and optimize the advice participants get. Nearly twenty years of research in usability testing show that involving users in every step of the design process, in an iterative design cycle, is key to achieving usable interactive programs. Case managers will therefore participate in this process by conducting a series of timely and strategic usability tests. This "user-centered design" approach [ND86, Wal02] lets us eliminate design flaws and problems with usability. Usability testing takes into account (1) a description of intended goals, (2) a description of the components of the context of use, including the users, the tasks they are to perform, the equipment they are to use, and the physical, regulator, and social environment in which they work, and (3) target values of effectiveness, efficiency, and satisfaction.

4. Lay the groundwork for further studies. Our proposed research has a very important long-term extension, not part of this proposal but enabled by its success: to see whether our software can help participants navigate the bureaucracy better. Recent and ongoing research on adults with low education and income levels shows that their decisions about whether to participate in education and work training programs hinge on complex individual assessments of the personal benefits and costs of such participation [JHGA00, Kuc99, Hul97]. They consider factors such as life experience in education, work and family settings; availability of education and work in their locale; personal responsibility to dependents; mobility and socio-economic safety networks; personal assessment of learning and retraining ability; comfort interacting with agency and institution officials (such as social workers, teachers, case managers, enforcement agents); and when known, social welfare regulations, benefits and liabilities. Research indicates that individuals make reasoned, reasonable choices based on what they know, albeit not decisions that Welfare to Work and adult education professionals might recommend [JHGA00, CN99]. They struggle to maintain a sense of dignity in and control over their lives. They often make choices with limited contact with social-service providers and with limited knowledge of social-welfare program availability and regulations. We would like to find answers to the following research questions: (1) Does access to an online planner assist these individuals in making choices more fully informed by Welfare to Work options? (2) Will people who avoid Welfare to Work planning due to reluctance to rely on case managers be willing to approach such planning through an online planner? (3) Will using such a planner help them engage Welfare to Work programs in ways that enhance their self-efficacy and thereby improve their likelihood of setting goals that they can achieve and that might move them to greater economic self-sufficiency and societal participation? Once the present project is sufficiently advanced we intend to seek separate funding for comprehensive studies of these questions.

C.5 Significance of the research, relevance to ITR program, and broader impacts

Our proposed research will create a complex software package that will augment the capabilities of human planners as they perform societally important functions. This package is a novel use of IT that benefits society at large.

Issues and ideas: The work is fundamental. This proposal addresses fundamental research in IT, specifically how to model complex, real-world planning applications with a combination of qualitative (logic-based) and quantitative (probabilistic) techniques. Our work will push the IT frontiers beyond efforts that fall within the scope of industry investments.

The work is novel. Current work in this field includes software for scheduling exams, offering academic advising solely on the basis of requirements, intelligent tutoring and testing, and diagnosing learning disabilities. No research, as far as we know, applies decision-theoretic planning to academic advising or to government programs like Welfare to Work.

The work involves storing complex data. We propose a unified software package that uses databases to represent probabilities and to support planning with constraints, utilities, and costs.

The work is an unusual collaboration. Our application of computer science to social-work problems represents a collaboration across widely separated academic domains: Computer Science (AI and Databases), Educational Policy Studies, and Instruction and Curriculum. Within computer science, our proposal involves collaboration among the areas of databases, planning under uncertainty, and constraint solving. This proposal is thus multidisciplinary at two different scales.

The work advances social and scientific goals. Our research will shed light on how to fuse probabilistic and logic-based modeling, how to populate the resulting models, and how to employ the populated models in planning. Our software will generate and compare plans that take into account the constraints of the application along with personal preferences of the advisees and probabilistic predictions of the outcomes of actions. Our software will improve the productivity of human planners, and, we hope, the quality of the plans they suggest.

The work can be applied to other important problems. Our research will point the way for a host of other real-world planning applications, such as financial planning, insurance advising, and city planning.

Research challenges: This proposal fits in the IT universe under the rubric of modeling human behavior, interactions, and cognitive processes. It addresses the following research challenges.

- It will extend our capability to process and manage planning information on an unprecedented scale of complexity.
- It will expand our capability to respond through IT to new opportunities. Our work will show how to build software for automating planning and model building in other complex, real-world applications. It will therefore lower the lag time between concept and implementation for those new applications.
- It will provide new computational methods and tools to model social and behavioral phenomena. The applications we will study involve a complex interplay of states, actions, constraints, utilities, costs, and probabilistic predictions of outcomes.
- It will improve our ability to understand and model the behavior of complex systems. Although it is not the thrust of our research, our models could be used to discover emergent behaviors, and this information could be useful to policy designers.
- It will integrate advances in IT into research in science in a way that will enable novel insights about the social world that we inhabit.
- It will develop a technology that enhances human abilities and efficiency. In particular, our automated planners will enhance the abilities of human advisors to generate high-quality plans quickly.

Integration of research and education: Our proposal addresses several broader issues. (1) Our research into the Welfare to Work application should have a positive long-term effect of improving the chances for advisees to succeed, leading to increased diversity in the general workforce. (2) Our research team fuses different communities both within computer science and between computer science and educational policy. (3) Our proposal advances the IT capability of the broader workforce, placing novel IT techniques in the hands of human advisors with no prior IT experience. (4) It points the way to a longer-term collaboration between the social sciences and computer science in the areas of modeling. (5) This interdisciplinary research collaboration will give us opportunities to train students in Computer Science and the School of Education in the techniques of both areas.

Societal benefits: Welfare to Work case managers are overworked, underpaid, and often undertrained. It is both costly and time-consuming to retrain case managers whenever domain details change. Our software has great potential value to case managers, generating plans tailored to the individual advisee while satisfying legal and budgetary constraints. Changes in advisee history, advisee preferences, and legal or budgetary constraints lead to modifications in the model; these modifications propagate to produce updated plans.

The Welfare to Work application needs to balance many and often conflicting individual and institutional priorities and utilities. For instance, an advisee may assign a high utility to not working, but the government assigns a high utility in placing the advisees in a job. Our model allows us to input the relative importance of satisfying individual and institutional goals and lets the software try to satisfy both as far as possible within the balancing constraints.

In summary, automating the planning process for the advisees and service providers will enable more efficient delivery of services, more effective use of services, and a higher success rate measured by all relevant criteria.

C.6 Personnel

The project team consists of 5 computer scientists and 2 social scientists. This composition reflects the character of the project: fundamental computer science research verified in a societally important area.

The expertise of team members covers all major areas required for the successful completion of the project:

1. Decision-theoretic planning: Dr. Goldsmith has studied the complexity of planning with Markov decision processes [MGA97, MGLA00, LGM01].
2. Logic-based artificial intelligence, computational knowledge representation and constraints: Drs. Marek and Truszczyński have developed theoretical foundations of nonmonotonic knowledge representation formalisms [MT93, DMT00a, DMT03], proposed the paradigm of answer-set programming [MT99], and developed successful implementations of answer-set programming systems [CMMT99, ET00, ET01b].
3. Logic-based AI: Dr. Dekhtyar has developed logic programming languages for storing probabilistic and temporal information [DDS99a, DS00, DDS99b].
4. Probabilistic information elicitation and fusion: Drs. Dekhtyar and Goldsmith have studied probabilistic databases and models [DRS01, DGH01, GZD03, DG02, DGZ03], fusion of probabilistic information [DGP01], and elicitation of preferences [RHGD02].
5. Constraint languages and software: Dr. Finkel has studied languages to support constraint elicitation [FMT01, FMT02].
6. Expert knowledge elicitation: Dr. Goldstein has studied qualitative and quantitative methods of knowledge elicitation in the social sciences [GB92b, GC91].
7. Human-computer interface evaluation: Dr. Mazur has extensive background both in user-centered interface design and social work [MM01, JML02].
8. Advisor-advisee interaction: Drs. Goldstein and Mazur have studied the interactions between providers and clients in the contexts of education and social services [GAJH02, GB92a].

C.8 Management plan (two additional pages over 15-page limit allowed)

Computing infrastructure and support for the project will be provided by the *Laboratory for Artificial Intelligence and Advanced Databases* (**LAIAD**), one of the research facilities in the Department of Computer Science, headed by Drs. Dekhtyar and Truszczyński, offers 1000 square feet of laboratory space and 17 dual-boot (Linux, Windows 2000) PCs. The laboratory also has two 100 GB RAID Arrays. Three other PCs with processor speeds of 3.0 MHz are currently on order for the Laboratory (funded by the College of Engineering funds). A technical staff member in the Computer Science Department will be responsible for ensuring reliable operation of this facility. This technician will be available for 4 hrs/week to maintain computing resources for the project; our budget requests funding to cover that expense.

The Department of Computer Science will also provide seminar and conference rooms (in its new research building) to facilitate group meetings involving researchers and students from Computer Science, the Department of Educational Policy Studies and the Department of Instruction and Curriculum.

Dr. Goldsmith, the Principal Investigator, will have overall responsibility for managing the project, its budget and its research team. Individual members of the team will have the following responsibilities:

1. Dr. Goldsmith: project director; research in decision-theoretic planning, facilitator of research interactions within computer science group and cross-disciplinary work involving the Computer Science and Educational Policy Studies Departments
2. Dr. Dekhtyar: research in probabilistic databases, research and development of corresponding software support tools
3. Dr. Goldstein: cross-disciplinary research on Welfare to Work; contacts with external organizations involved in the project
4. Dr. Truszczyński: research on constraint processing and integrating constraints in quantitative approaches
5. Dr. Finkel: research and development in the area of constraint languages and corresponding software support tools
6. Dr. Marek: research on integrating logic-based knowledge-representation formalisms with quantitative approaches
7. Dr. Mazur: evaluating the planning software in the Welfare to Work application.

We will manage the project through the following administrative steps:

1. Weekly meetings held by each group forming the team: quantitative AI, led by Dr. Goldsmith, logic-based AI, led by Drs. Marek and Truszczyński, probabilistic databases, led by Dr. Dekhtyar, constraint languages, led by Dr. Finkel, and the Welfare to Work application, led by Dr. Goldstein.
2. Bi-weekly project meetings led by the PI and involving all personnel (researchers and students); discussion of current issues in research, with a goal to facilitate integration of quantitative and qualitative approaches and to promote cross-disciplinary collaborations.
3. Bi-weekly meetings led by Drs. Goldsmith, Goldstein, and Mazur involving personnel directly involved in activities related to the Welfare to Work application.
4. Quarterly project meetings led by the PI and involving all senior research personnel to evaluate progress towards long-range objectives, to revise research approaches, if necessary, and to review budgets
5. Annual mini-workshops to present research results and software to the University of Kentucky community and to prepare the annual NSF report.