When Domains Require Modeling Adaptations

Krol Kevin Mathias*, **Cynthia Isenhour**[†], **Alex Dekhtyar***, **Judy Goldsmith***, **Beth Goldstein**[‡]
{kevin9,Cynthia.Isenhour}@uky.edu, {dekhtyar,goldsmit}@cs.uky.edu, bethg@uky.edu
University of Kentucky, Lexington, KY

Abstract

The project described in this paper originated with an observation by the AI group at the University of Kentucky, that, individually, stochastic planning and constraint satisfaction are well-studied topics that resulted in efficient software, but stochastic planning in the presence of constraints on the domains and actions is an open area of investigation.

We were interested in an advising scenario, and chose the US social welfare system, a.k.a. "Welfare to Work" as our test domain. This required computer scientists to learn more than expected about social science as well as the local welfare system. This paper discusses the discipline specific assumptions we brought to this project, and how they served as impediments to research. We also show how the different perspectives have sparked new ideas in knowledge elicitation.

1 Introduction

When you give advice to someone, you have to assume that the outcome of that advice is not determined. The advisee might act on your advice, or might ignore it. If she acts on it, her actions may succeed, with a variety of possible effects, or may fail, with equally undetermined effects. At best, you can put probabilities on possible outcomes—if you have statistically significant experience with the individual or the circumstances.

We rarely give advice in a knowledge vacuum. We are affected by our own biases and by what we know of the advisee's preferences, as well as by any known predictors of success. For instance, we might recommend that an advanced and bright undergraduate take a graphical models course, while a student who has repeatedly flunked precalculus should avoid that course. Furthermore, we attempt to avoid advising the impossible, whether it is for a student to take two courses meeting at the same time in different locations, or an indigent friend to buy his fiancee a diamond ring.

In short, we can model advice-giving in terms of *factored Markov decision processes (MDPs)*, where decision variables represent the advisee and actions represent what we advise them to do. However, there is a piece missing from the MDP model: *constraints*. There is no straightforward way to represent, for instance, that a student could take the graphical models course or underwater basket weaving, but cannot take both; we could code pairs of what we call *elementary actions* as the MDP actions, but we are unlikely to want to work with an MDP that has an action for every subset of the elementary actions. Quite the contrary, the attractiveness of using MDPs for modelling advising lies in our ability to represent them in a factored form using dynamic Bayesian networks. In addition, as we just argued, we need to add something to the MDP formalism, namely explicit constraints.

When Judy Goldsmith was discussing her interest in planning under uncertainty with constraints with her friend Beth Goldstein, Beth said, "If you want planning under uncertainty with constraints, you should look at the welfare system."

The current US social welfare system is intended to move recipients into jobs if at all possible. A recipient, or client, meets with an assigned case manager, who negotiates a contract between them. The client will participate in certain activities, and the case manager will authorize support in various forms, including healthcare, childcare, transportation, school or training, and a stipend. However, each client has a 60-month lifetime limit on services.

* Department of Computer Science.
† Department of Anthropology.
‡ Department of Educational Policy.

It seemed to Judy that the welfare domain was no more complicated than academic advising, and probably much more fundable. She was correct on one out of two assumptions.[1] There were many assumptions that she did not make explicit, much less question. She assumed that:

- welfare case managers would be happy to supply information and would want decision-support software;
- her computer science colleagues knew how to build appropriate knowledge-elicitation software;
- her social science colleagues would quickly grasp how computer scientists model the world and would be able (and willing to) reason in the same terms;
- case managers would be able to quickly grasp the notion of dynamic Bayesian networks, and would be able to supply attributes, dependencies, and conditional probabilities;
- she need only explain the problem to the constraint-satisfaction folks and they would be able to use their constraint solvers to speed up factored MDP solvers;
- managing a group of seven professors and a varying number of students would be straightforward, and everyone would work productively with no supervision;
- by the end of four years, we would have models and fully integrated software to offer the case managers, who would have the freedom to choose to use that software.

It is not uncommon for scientific research to come across hurdles that have little to do with the technical content of the study. It is, however, much less common for the technical content of the research to be significantly influenced by those hurdles.

The contributions of this paper are twofold. In Section 2.2 we discuss the story of our project and how the above mentioned implicit assumptions came crashing down one after another in the course of the project. More importantly, we describe the adjustments in our approach to conducting the research, and outline the overall lessons learned and successes achieved. In particular, (and this is the second contribution of the paper), we concentrate on the process of model elicitation employed during the project. In Section 3 we describe the evolution of our approach to data elicitation from case managers, the evolution of the data model under the influence of case managers and the social scientists involved in the project, the software development process and the eventual model elicitation experiment. This paper is intended as a companion to [6]. Where [6] concentrates on the technical aspects of our research, this paper addresses the evolution of our approach, and the lessons learned thus far from the collaboration between the computer scientists and the social scientists (Section 4).

[1] This work is supported by the NSF grant ITR-0325063.

2 Decision-theoretic Planning for Welfare-to-Work

The project described in this paper originated with an observation by the AI group at the University of Kentucky, that, individually, stochastic planning and constraint satisfaction are well-studied topics that resulted in efficient software, but stochastic planning in the presence of constraints on the domains and actions is an open area of investigation.

There are certainly MDP solvers that handle constraints. These include linear programming-based solvers that can include any linear constraints (see, for example, [4] for an example solver for factored MDPs). There are solvers which directly convert the MDP and its constraints into a constraint satisfaction problem (see [3] for an example). However, we expect that solvers for MDPs with constraints will be a growth area in AI in the next few years.

Note, however, that there is a competing notion of MDPs with constraints, typified by [1]. In these models, there are indeed constraints, but what is constrained is the range of acceptable cost/reward functions. This might be of interest to us, but is not our primary focus.

As a domain for stochastic planning with constraints we have considered advising settings. In such a setting one human agent, the advisor, is tasked with suggesting to another human agent, the advisee, a plan of actions. In coming up with a long-term plan, the advisor has to base her decisions on three sets of criteria: (a) the perceived stochastic effects of the actions taken on the chances of succeeding in other actions; (b) constraints on which actions and action combinations can be taken under which circumstances, and (c) the preferences stated by the advisee.

In Section 2.1 we discuss our approach to modelling situations when stochastic planning with constraints is needed. In Section 2.2 we discuss the Welfare-to-Work application in more detail.

2.1 Planning with uncertainty and constraints

The key assumption behind our work is that in the general advising setting described above, the advisor considers that taking an action in a state has stochastic effect. That is, the results of taking an action can be described as a probability distribution over a set of possible new states. In our original setting, academic advising, this made perfect sense: taking a course in Databases could lead to an "A", a "B", a "C", a "D" or an "F" in the course. A current student transcript could suggest the likelihood of a student earning each grade: a student with a 4.0 GPA is more likely to earn an "A" in the course than any other grade, whereas a student with a "C" in data structures is more likely to earn a "C" in databases.

The second key consideration in the frameworks we consider is the fact that our actions and states are factored. The advisor considers the advisee's state to consist of a number of $\langle Name, Value\rangle$ pairs describing atomic "pieces" of information about the advisee, such as individual grades in courses already taken. The advisor considers a list of possible *atomic* actions that can be taken by an advisee (such as taking a course), and can suggest any subset.

The constraints the advisor has to consider deal with both states and actions. Certain combinations of actions may be unavailable/prohibited (e.g., taking two classes that meet at the same time). Certain states are not allowed (e.g., being an honors student and having a GPA of 2.5). Some actions may be prohibited in some states (e.g., taking Advanced Databases, if the grade for Databases is "F").

The final aspect of our planning framework is the origin of the goal function. Instead of considering a specific definition of a "successful plan" (e.g., plan that maximizes the probability of getting the highest possible GPA), we assumed that each advisee comes with her own goals, expressed in a form of factored, disjoint, and potentially contradictory preferences over the domain variables and their available properties (represented in the model as meta-information). For example, one student may have as a goal a fast-track graduation, and be willing to lower her GPA in order to get a degree as soon as possible. Another student may want to obtain in-depth knowledge in the areas of Databases and AI, and thus would be willing to wait for an extra semester or two, as long as she gets to attend the classes she wants when they are taught by her favorite professors. A third student might be ambivalent about specific classes and their effect on his GPA, but request only mid-afternoon classes, due to a work schedule or a parallel career as an aspiring rock musician.

The factored nature of our framework led us to adopt Bayesian networks as the representation model for action influences on states. The nodes in the Bayesian network, the random variables of the domain, represent various advisee characteristics that can be acquired and changed stochastically (grades in courses, interests in topics, and so on). Additionally, non-stochastic meta-information is associated with both the advisee (e.g., major, year in school) and the action (time, instructor of the course, location). The meta information forms the domain over which advisees can express conditional preferences (e.g. "I would like very much to take AI if Goldsmith is teaching it, but I would prefer not to take her Theory course, especially if it is taught in the morning").

For this sort of uncertainty, dynamic Bayesian networks work well.

2.2 Work in the WtW domain: Barriers and challenges

As mentioned in Section 1, at its outset, planning for Welfare-to-Work clients appeared to be similar to planning in the academic advising setting. Indeed, the Welfare-to-Work system gives case managers the ability to advise combinations of actions such as participation in different services, training classes, volunteer work, etc. Each such action has the potential to change the client's state. These changes are uncertain and can be modelled stochastically. The action space and the current information available about the client are factored.

In addition to this, the Welfare-to-Work system operates under a wide array of federal, state and local rules and regulations. These supply a rich set of constraints, from the 60-month limit on benefits over an individual's lifetime to soft constraints on "countable" activities (those that go toward meeting the case manager's and the agency's quotas) and "allowable" activities. There are also logistical constraints. For instance, a client who relies on public transportation must begin and end activities while public transit is running, and must be able to reach those activities.

Client preferences also play a role in determining courses of action. Even if certain activities may be beneficial to a client, she may want to forgo them (e.g., a client has the potential for a career in health care, but has a strong aversion to blood).

All of the above suggested to the computer scientists that the theoretical model developed for the academic advising domain would be immediately applicable to the Welfare-to-Work domain. The key difference between these two domains seemed to be that all Computer Science faculty working on the project were well-acquainted with academic advising, and had considerable expertise in it, whereas none were in position to consider themselves experts in the Welfare-to-Work domain. To compensate for this, the AI group teamed up with social scientists who had experience working (on unrelated projects) with Welfare-to-Work system personnel.

What the AI group did not anticipate is that these regulations and logistics change frequently. Services become available or unavailable; laws change; case manager and agency quotas may fall more heavily on clients at the end of cycles. Case managers begin negotiations with incomplete information about clients. Client preferences change as clients gain information and experience.

In addition, the assumptions made at the beginning of the project, and outlined in Section 1 turned into barriers. In particular:

- Case managers worry about being put out of work by our software;
- graphical user interfaces designed by computer science students and faculty on their own turned out to be hard to use for non-computer scientists;
- social scientists approached various project-related issues, from theoretical constructs to practical activities, with a completely different mindset than computer scientists did, and it took a long time for the computer scientists to recognize this;
- dynamic Bayesian networks did not provide case managers with enough intuition; case managers, for the most part, reason from narratives, not quantitatively, and exhibit little desire to abstract from the narratives in order to capture the common aspects of their reasoning and decision-making in different cases;
- Markov Decision Processes and constraints go together like garlic and chocolate;
- computer scientists and social scientists brought different expectations about collaboration, so management of the project resembled, at times, herding cats;
- at the close of the third year of the project, we have built strawman models, are in the process of building our first "real" model, diverse software is just starting to be integrated, and getting *any* software on the case managers' computers requires an act of State Legislature.

These hurdles have significantly shaped our project, forcing it to adapt to unforeseen realities. In the section to follow, we describe how our perceptions of Bayesian models were significantly altered based on the knowledge we received from both the case managers and social scientists.

3 Elicitation of Models

Elicitation of information for construction of Bayesian models of advising in the WtW domain is central to this project. Originally, computer scientists proposed to represent activities (actions) a WtW client can take as two-phase Bayesian network (a 2TBN or DBN) [2]. Each activity, described as a DBN fragment, showed how various client characteristics were likely to change, based on their current state and the WtW client completing the action.

From the beginning of the project, social scientists worked with a group of case managers. Through multiple interviews, they have established and conveyed to the computer scientists the main operational procedures and key regulations that guide the WtW program. They have also discussed with the case managers their *modus operandi*, trying to establish how, in general, case managers assessed likelihood of their clients' success in different activities.

One of the surprises for the computer scientists was the rejection by the social scientists of the two-layer DBN fragment as the model of actions. When case managers were asked questions about client characteristics that influence their decision to recommend a specific action, as well as the expected change of client characteristics, case managers refused to answer and motivated their refusal in two ways. First, case managers insisted that they would not make generalizations and discuss "generic" clients. Their vivid experiences with clients made it hard to hypothesize about their actions in the presence of a client described by a list of characteristics. Asking "what would you advice to a 24-year old mother of two who lives in an apartment complex, lacks transportation, has a high-school diploma, but has no work history and an has a history of alcohol abuse?" turned out to be a wrong type of question—too impersonal for case managers to be able to give answers.

In general, case managers agreed that the actions their clients take affect their "state". What they did not agree with was the idea that the mere act of taking an action changes the state, as implied by the DBN model structure. The missing piece, in the opinion of the case managers was the *result of the action*, i.e., success or failure. In a sense, a clear outcome from pre-elicitation interviews with case managers was the necessity of representing the success of an activity explicitly.

The goals and objectives of case managers are tied directly to helping a client succeed in a given action. According to case managers, a client's success—or lack thereof—in an activity has a profound positive impact on the client's state. This in turn affects the client's likelihood of sucess in future actions. The DBN model did not represent the transformation between two client states based on the *explicit outcome* of the client's participation in an activity.

To address the concerns of case managers and to facilitate knowledge elicitation from them, computer and social scientists jointly developed a new class of stochastic models, which we call *bowtie action fragments* [6]. The new model introduced, for each activity, a *success node*, a random variable explicitly quantifying the client's performance in, or level of success in completing, the activity. The success node became the central node of the bowtie models. Client characteristics from the current state influence the success node. The success node, in turn, influences the client characteristics at the next client state, upon completion of the activity.

Figure 1 represents the originally considered DBN model and the bowtie model for the action "Volunteer Placement (VOP)". In this action, the client participates in volunteer work relevant to the client's job goals. The bowtie model represents the action framework in the WtW scenario. However, some influences in the bowtie model are not explicitly represented. In particular, *each output node in an action fragment is not only influenced by the action success node but also by the node representing the value of this characteristic prior to the action being taken.* The first influence is represented explicitly in the bowtie diagram, but the second is implicit (represented by dotted arcs) in the bowtie diagram.

The elicitation of case manager knowledge was done in three stages: (i) a manual pilot study carried out by the social scientists, (ii) design and implementation of elicitation software, and (iii) software-directed elicitation. We briefly outline these stages below.

First, our team of social scientists conducted a pilot study to test the bowtie elicitation methodology. In the pilot study, welfare case managers were asked to free list client characteristics which would affect the client's likelihood of success in the action "Get GED", which includes attending preparatory classes and eventually taking the Graduate Equivalency Degree (GED) test.

The welfare case managers listed nearly 30 characteristics that would affect the client's likelihood of success in the action "Get GED". These 30 characteristics were then evaluated and ranked by small groups. The five client characteristics cited by case managers as most important were weighted, thus forming the input structure of the bowtie model for the "Get GED" action.

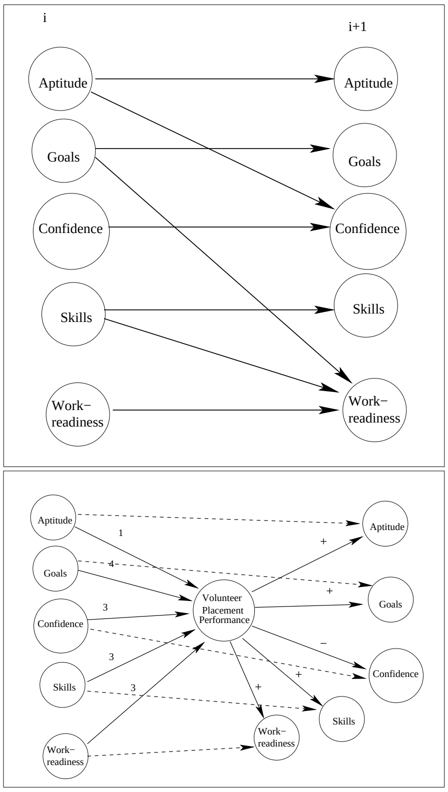

Figure 1: DBN model and bowtie model for the action "Volunteer Placement (VOP)".

In order to get more precise information on the action models, we decided to merge the list of 200 client characteristics generated in the pilot study and extracted from previous expert interviews into a more manageable list of approximately 50 variables. We also loosely categorized these variables as education-related, work-related, and personal characteristics. Within each category, we tentatively outlined subcategories. The list of most often considered characteristics and their classification into categories and subcategories is shown in Figure 2.

The elicitation methodology was verified and the experiment was replicated for the additional fifteen actions in which a welfare client can participate (Table 1). As part of our study, we elicited a total of 16 actions from case managers. Out of these, "Get GED" was used in the pilot study as a "training example" and then excluded from further elicitation procedures.

Name	Abbrevn.	Name	Abbrevn.
Job & Post-Sec. Edu.	JSE	English Second Language	ESL
Short Term Training	JST	Volunteer Placement	VOP
Voc. Rehabilitation	RHB	Job Readiness Class	JRA
On Job Training	OJT	Literacy Training	LIT
College	COL	Community Service	COM
Adult Basic Edu.	ABE	Community College	CCO
High School	HSC	Voc. School	VOC
Group Job Search	GJS		

Table 1: The list of 15 elicited actions.

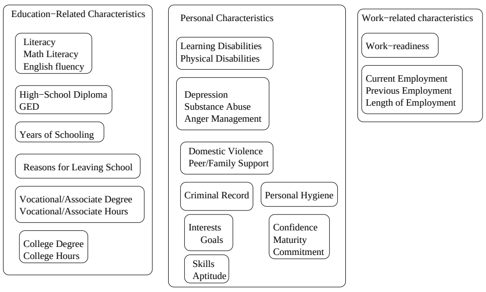

Figure 2: Client characteristics used in the experiments.

A High Level Elicitor (HELL), the special-purpose elicitation software developed by our team for eliciting bowtie action models, was utilized to replicate this methodology. This software was built with the combined effort of the social and computer scientists.

The High Level Elicitor was designed to obtain bowtie models by eliciting the input of 18 participating case managers. Each manager was assigned 5 of the 15 actions from Table 1. We loosely categorized the actions as education-related or work-related. Each case manager was provided with an equal opportunity to elicit information on work-related and education-related actions.

For each of their five actions, they were asked to complete a three-step process. In the first step, case mangers were asked to pick, from a list of 50, the top five client characteristics that would affect a client's participation in the given action. In the second step, the case managers were asked to assign a weight (from a scale of 1–4) to each client characteristic selected in step one. This helped us to determine the relative importance of the five selected client characteristics. In the third step, the case managers elicited the output nodes for the action, by indicating which of the client characteristics were most likely to change positively or negatively as the result of the completion of the assigned action. For example, case managers reported that, the client would experience an improvement in Aptitude, Goals, Skills, and Work-readiness, upon successful completion of the action Volunteer Placement (VOP).

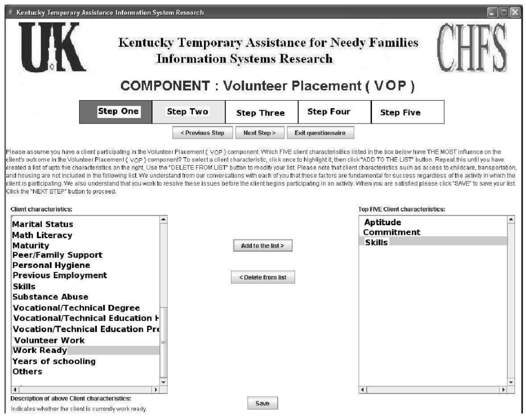

Figure 3: Step 1 of the elicitation process for the action fragment "Volunteer Placement (VOP)".

4 Lessons Learned

To computer scientists, a process is a thread, whereas to social scientists, process is a verb. We entered this project knowing—as individuals—how collaborative research was conducted. Unfortunately, we had no *common* model of the collaborative process. The computer scientists assumed that they could pick apart the technical requirements, develop solutions in parallel, then integrate. The social scientists expected a leader or coordinator who understood methods and goals and kept the separate threads synchronized. As those threads got out of sync, the social scientists became bewildered and frustrated. Their frustration surprised some of the computer scientists.

The team learned the importance of having social and computer scientists working together on the software development life cycle (SDLC) of software and solutions for the WtW project. Social scientists were able to translate the needs of computer scientists into a language that made sense to the case managers given their own perspectives, needs, and interests and vice versa. This helped immensely in the requirements gathering phase of the SDLC of the model building/elicitation software, i.e., HELL. It was evident to the team that the intended users, the case managers, were more likely to be responsive to software programs that directly addressed their needs and desires. Any software that needed case managers' participation had to be built around their reality; theoretical models that work well with computer science research were not sufficient. Applied anthropology's emphasis on user-centered development programs led the ethnographic team to frequently caution the computer scientists to not go too far with their assumptions and objectives until the case mangers were involved in both the definition of the problem and the process of imagining possible solutions. We also learned key issues like usability, cognitive overload, and information non-clutter that need to be considered while developing research software.

The social scientists made the computer scientists aware that their relationship with case mangers in the WtW project was very different from the relationship between a development team and clients in a traditional software development setting. The team learned that, in order to build software solutions that would be useable by case managers, it was very important to affirm the case managers' professionalism. The case managers also had to be reassured that their participation in any of the research experiments (for e.g. model elicitation) was not a waste of time but was giving them something valuable in return.

The team also learned to deal with challenges that came up due to interdisciplinary work. Some of the great challenges emerged not in understanding what esoteric terms like Bayesian network mean, but rather from seemingly simple and common terms such as "value", "variable", "state", and "utility" [7]. While each of these words are used commonly in English language, the team found that they have dangerously subtle differences in implication and connotation depending on the discipline of the team member. Even subtly different usages of these terms meant that few members of the team were unclear about the software being designed and about the type of and format for the information required. Our collective deconstruction of the terms also forced members of the team, often from the same discipline, to rethink assumptions.

Social scientists typically work in a relatively inductive and empirical fashion as compared to computer scientists. Computer science comes out of a much more positivist tradition which places a lot of emphasis on deductive and generalized reasoning. This led to challenges while building the HELL software. The social scientist placed more emphasis on specific client profiles for eliciting information about different actions in the WtW, whereas the computer scientists wanted to build more generalized (abstract) models representing the actions. The team learned to merge these contradictory ideas into a single requirement during the development of the elicitation software.

5 Conclusions

While working with social scientists and experts can complicate matters in the development stages, it ultimately will result in a better package, more suited to the needs and desires of the end users.

Reality is complicated, While great things can be done via abstraction, it takes time to figure out how to abstract data in ways that are both valid and reliable. Meanwhile, because building a correct and complete model is a slow process, we have developed a simplified model on which to test our solvers. We are making that available through a parallel submission to this workshop [5]. However, working with empirical rather than stand-in data is initially more complicated but also results in more accurate models and plans.

When working in an interdisciplinary team it helps us to be open and flexible to different ideas and paradigms. By keeping an open mind and listening to the experts, the computer scientists were able to recognize and embrace the emergence of a new Bayesian model which more closely resembled the case managers' reality.

6 Acknowledgments

This work was partly supported by NSF grant ITR-0325063. The opinions expressed here are those of the authors and do not represent the Foundation, the University, or any social welfare offices. We also thank Russell Almond for enabling communication between the computer scientists and social scientists about bowties.

References

[1] E. Altman. *Constrained Markov Decision Processes.* Chapman and Hall/CRC, 1999.

[2] Craig Boutilier, Thomas Dean, and Steve Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *J. Artif. Intell. Res. (JAIR)*, 11:1–94, 1999.

[3] Rina Dechter and Robert Mateescu. Mixtures of deterministic-probabilistic networks and their AND/OR search space. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, 2004.

[4] Dmitri A. Dolgov and Edmund H. Durfee. Symmetric primal-dual approximate linear programming for factored MDPs. In *Proceedings of the Ninth International Symposiums on Artificial Intelligence and Mathematics (AI&M 2006)*, Florida, January 2006.

[5] Kendra Renee Gehlbach, Brandon Laracuente, Cynthia Isenhour, Judy Goldsmith, Beth Goldstein, and Miroslaw Truszczyński. A benchmark model for decision-theoretic planning with constraints, 2006.

[6] Krol Kevin Mathias, Cynthia Isenhour, Alex Dekhtyar, Judy Goldsmith, and Beth Goldstein. Eliciting and combining influence diagrams: Tying many bowties together. Technical Report TR 453-06, University of Kentucky, March 2006.

[7] Joan Mazur, Beth Goldstein, and Judy Goldsmith. Social construction of technology in the welfare to work project. In *Proceedings of the Second Bayesian Modeling Applications Workshop (UAI 04)*, 2004.