Learning User Preferences by Satisfying Knowledge Goals.

Tamara Babaian
CIS Dept, Bentley College
Waltham, MA 02452

Abstract

We present a novel approach to software customization that uses a logic-based PSIPLAN language to represent personal information and specify the task of personalization as a set of knowledge goals. The method uses automated planning and execution of plans as the means to satisfy personalization goals. Thus learning is carried as a process of achieving information goals via reasoning and executing information gathering plans that may include direct user querying. The advantages of the approach are discussed and illustrated by examples from the Writer's Aid application.

Introduction and Motivation

It is acknowledged (Manber, Patel, & Robison 2000) that even though users can benefit greatly from personalization they often do not use it. Modern systems often include tens if not hundreds of configurable options, and it is merely impossible for the user to specify the personal preferences on each of these options at installation time, and hence the systems are usually installed with a set of default settings. However, even experienced users often do not engage in further customization for various reasons including their unawareness on the potential benefits of personalization and thus the lack of motivation. This problem is at least partially due to the fact that the customization mechanisms often appear outside of the context of tasks in which customizable options play a significant role and therefore the users often cannot see the potential advantage of choosing a custom setting over the default.

On the other hand, from the system designer's standpoint as Pednault (Pednault 2000) observes the underlying representation of "the human-side and the technology-side" of the personalization task is the key, however representations currently in use at times lack flexibility to be easily adjustable and reusable.

In this paper we present an approach to personalization that provides a solution to the problems above by encoding personal information and the task of personalization in a declarative, and thus flexible and reusable manner, and also creates a mechanism by which customization can be performed gradually within the context of a particular system action and occur right before any customizable option is used for the first time.

We distinguish two phases of personalization: initial, that occurs prior to using the system for the first time, and further customization. The initial personalization occurs at the installation time and is carried out semi-automatically as a process of satisfying knowledge goals, i.e. goals of knowing the new user's personal preferences. All collected personalization information is stored in a knowledge base.

Further personalization occurs in the process of the system working on a user's request. System actions that use customizable options are tagged with preconditions that require knowledge of the user's preferences with respect to these options. Before an action is executed the system has to establish that the precondition holds, i.e. the personal information is available.

The method relies on automatic reasoning to determine what personal information can be inferred from the already existing knowledge base and what information is still missing. Automated planning is then used to construct and execute a plan that achieves the knowledge goal by information gathering (a.k.a. sensing) actions. Some of the sensing actions involve directly querying the user by posting a question through a dialog box or other means.

This approach is illustrated in the next section on the examples from Writer's Aid application (Babaian, Grosz, & Shieber 2002). Section presents an overview of the representation language PSIPLAN on which this approach is based.

Examples from Writer's Aid

Writer's Aid (Babaian, Grosz, & Shieber 2002) is a system that works parallel with an author writing a document acting as a collaborative partner that carries on the tasks of identifying and inserting citation keys, autonomously finding and caching papers and associated bibliographic information from various online sources.

Writer's Aid's knowledge about the state of the world includes the description of the user's preferences and in stored in a knowledge base called State of Knowledge

(SOK). An automated reasoning and a partial order planning algorithm are used to determine a course of action to satisfy the user's requests for bibliographic information. Each action is described via preconditions that must be true prior to executing the action, and effects that the action brings about.

Personalization in Writer's Aid consists of the *initial tune-up* of the system to the user's parameters and the dynamic personalization that occurs while Writer's Aid works on accomplishing a user-posted goal and identifies a need for information.

Initial tune-up occurs at the time of installation. The goal of the initial tune-up is to establish and enter into the system certain user-specific parameters, such as, for example, the user's own locally stored bibliographic collections, his preferred on-line bibliographies, etc.

To direct the system to collect the data about location of local bibliographies of a new user, Ed, it is sufficient to post the following goal on the list of goals to be accomplished during the tune-up

> g_1*: For each file know whether it is a BibTex file owned by Ed.*

This goal is expressed with a *KnowWhether* proposition of PSIPLAN. KnowingWhether p, or $KW(p)$ for a ground proposition p means knowing the truth value of p. Thus $KW(p)$ means knowing that p is true or knowing p is false and is is equivalent to $KW(\neg p)$. The goal g_1 is expressed by proposition $[KW(Bibtex(?f) \wedge Owns(Ed, ?f))]$ which (as explained in the next section) represents all possible instantiations of the $KW(Bibtex(?f) \wedge Owns(Ed, ?f))$ for all values of variable $?f$.

To satisfy this goal Writer's Aid will try and generate a plan which accomplishes and thus provides Writer's Aid with access to the user's personal bibliographies. The solution plan in this case would consist of a single sensing action of searching for local BibTex files, the effect of which entails g_1.

The goal of this example is satisfied by a single sensing action. Sometimes the knowledge goal is entailed by the existing knowledge, or a combination of existing knowledge and sensing. For example, suppose that Writer's Aid's knowledge base contained the following statement

> p_1*: There are no BibTex files owned by the user except possibly file ai.bib*

Given proposition p_1 and the goal g_1 Writer's Aid's reasoning module would determine that it only needs to find out whether file the file *ai.bib* is the user's own BibTex file and use an appropriate action to obtain this information.

This declarative approach to the initial customization as a set of knowledge goals separates personalization from the rest of the code, making personalization design very flexible and more easily adjustable.

Dynamically occuring personalization As mentioned, usually not all personalizable options can be obtained during the initial tune-up of the system. Further personalization in Writer's Aid takes place at the times when the knowledge of personal information becomes necessary to perform a system action.

For example, imagine the following scenario: Writer's Aid is working to locate a viewable version of a paper that the user requested. The plan for locating the paper includes an action of querying a known paper collection, namely ACM digital library. In order to avoid wasting time on searching collections of papers on subjects unrelated to the user's research field, this action contains a precondition that the paper collection be one of user's preferred collections:

Action-2: QuerySourceForPaper(source, paper)
Precondition: PreferredSource(source)
Effects: KW(HasPaper(source, paper))

Writer's Aid does not know if ACM digital library is the user's preferred bibliography, so it cannot establish the precondition unless it executes an action (namely Action-3 described below) of asking the user to obtain necessary information.

Action-3: AskUserAboutSource (source)
Precondition: QueriesPermitted()
Effects: KW(PreferredSource(source))

The user's response determines whether ACM digital library will be queried; it is also recorded in Writer's Aid knowledge base for future use.

Dynamic personalization occurs gradually, always within a context of a particular task, thus eliciting the user's input at the time it is used and providing the user with knowledge of how the personal information is being used by the system.

Overview of PSIPLAN

All of Writer's Aid's knowledge about the world is contained in the SOK database. This knowledge is assumed to be correct but incomplete.

Writer's Aid uses the PSIPLAN language (Babaian 2000) because this language enables efficient representation of the agent's incomplete knowledge about the domain consisting of an infinite number of unknown objects and knowledge goals, as well as efficient knowledge update after a sequence of actions. In addition, the reasoning about domain and knowledge from a set of domain propositions in PSIPLAN is sound, complete and takes only polynomial time in the size of the SOK database. (Babaian 2000) provides a complete description of the PSIPLAN language, its semantics, calculus and proofs of its properties mentioned in this paper.

Such precision in reasoning about the world in presence of the unknown bears a direct effect on nonredundancy of information gathering and thus is especially critical for a system that uses costly sensing operators or sensing operators that require user involvement. To guarantee non-redundancy of the sensing act, the planner only uses sensing where the truth value of a target condition is not already known.

Alternative representations that are capable of reasoning with universally quantified sentences are either intractable in the general case, or as the tractable LCW (locally closed world) representation (Etzioni, Golden, & Weld 1997), lack the desired completeness of reasoning.

PSIPLAN assumes infinite number of domain constants, and no other function symbols. PSIPLAN formulas are either *domain propositions* or *knowledge propositions*.

Domain propositions are either *atoms* or propositions called ψ*-forms* that are equivalent to universally quantified clauses with exceptions. The general form of a ψ-form is

$$[Q(\vec{x}) \text{ except } \{\sigma_1, \ldots, \sigma_n\}],$$

and it represents a possibly infinite set of ground negated clauses that are obtained by instantiating the formula $Q(\vec{x})$ called the *main form*, which is a disjunction of negated literals, in all possible ways (corresponding to all possible ground assignments on the variables in the main form), except for the instances specified by the substitutions σ_i, called the *exceptions*.

This combination of atoms and ψ-forms is necessary to describe situations as the following one, in which the agent knows that

> *The only bibliographies preferred by Ed are the digital library of the ACM, and maybe the ResearchIndex database.*

In PSIPLAN the example statement above is expressed by stating that

1. *ACM's digital library is a preferred bibliography*, which is represented by a ground atom:
$$a = PrefBib(ACM) \tag{1}$$
2. *Nothing is a preferred bibliography except for ACM and the ResearchIndex*, which is represented by the ψ-form:
$$\psi = [\neg PrefBib(b) \text{ except } \{\{b = ACM\}, \{b = RI\}\}] \tag{2}$$
Thus, ψ denotes all ground instances of the formula $\neg PrefBib(b)$ minus two exceptions: $\neg PrefBib(ACM)$ and $\neg PrefBib(RI)$ and is equivalent to the universally quantified predicate calculus formula
$$\forall b. \neg PrefBib(b) \vee (b = ACM) \vee (b = RI)$$

Note that ψ alone does not commit to the truth or falsity of its exception clauses, i.e. $\neg PrefBib(ACM)$ and $\neg PrefBib(RI)$, but from (1) and (2) we can conclude that the agent knows that ACM is a preferred bibliography, and nothing else is a preferred bibliography except for possibly the ResearchIndex.

Knowledge, or KW**-propositions** have form $KW(p)$, where p is a domain clause, represent "knowing p or knowing *not* p", i.e. that the value of a domain clause p is known without committing to a particular value. $KW(p)$ is semantically equivalent to $KW(\neg p)$. For example, $KW(PrefBib(ACM))$ represents knowing-whether ACM is a preferred bibliography.

Knowledge ψ-forms, similarly to domain ψ-forms, represent a conjunction of all ground instances of the main form. For example, a KW-ψ-form

$$\tilde{\psi} = [KW(PrefBib(b))] \tag{3}$$

represents a conjunction of all ground instances of the main form $KW(PrefBib(b))$, and thus the fact that the set of all preferred bibliographies is known.

Knowledge propositions in PSIPLAN are used to encode information goals, results of sensing actions, and ignorance about a domain proposition. For example, posted as a goal, $\tilde{\psi}$ requires knowing the value of each ground instance of $PrefBib(b)$, or in other words, knowing the set of preferred bibliographies. The effect of checking if RI is a preferred bibliography, is a knowledge proposition $KW(PrefBib(RI))$.

A negated kw-proposition $\neg KW(p)$ represents the agent's ignorance about p.

SOK (State Of Knowledge) database is a consistent set of PSIPLAN domain propositions. It represents the knowledge available to the system in the following way:

1. a domain proposition p is true in the world, if and only if $SOK \models p$,
2. furthermore, we make a Closed Know-Whether Assumption (**CKWA**) and assume that if $SOK \not\models KW(p)$ then the truth value of p is not known, i.e. $\neg KW(p)$

The set of possible worlds corresponding to this representation consists of all world states in which everything known to the agent is true, and only things known to the agent are guaranteed to be true. Such representation is sound and complete, due to soundness and completeness of reasoning about domain and knowledge propositions from a set of domain propositions in PSIPLAN. Importantly, the inference procedures also run in polynomial time and are fast, which bears directly on the speed of planning with PSIPLAN. PSIPLAN thus ensures precise and fast reasoning about knowledge and ignorance.

For example, given the goal of knowing the set of all preferred bibliographies (3), PSIPLAN reasoning will establish that, given atom a from (1) and ψ-form ψ from (2), the only thing left to find out is whether or not the ResearchIndex is a preferred bibliography. This computation is performed by PSIPLAN's special operation of *e-difference* or extended difference, denoted $\dot{-}$. For two sets of ground propositions $A \dot{-} B$ represents the subset of propositions in A that are not entailed by B. ψ entails knowledge of proposition $\neg PrefBib(b)$ for any value of b, except

$b = ACM$ and $b = RI$, and therefore ψ entails all propositions of $\tilde{\psi}$ except $KW(\neg PrefBib(ACM))$ and $KW(\neg PrefBib(RI))$. $KW(\neg PrefBib(ACM))$ is in turn entailed by $a = PrefBib(ACM)$, and thus

$$(\tilde{\psi} \dot{-} \psi) \dot{-} a = KW(\neg PrefBib(RI)).$$

E-difference is computed *without* expanding $\tilde{\psi}$ into a conjunction, using only unification and substitution operations.

PSIPLAN Actions PSIPLAN actions are represented by operator schemata, specifying each action's *argument list*, *preconditions* and *effects*.

PSIPLAN distinguishes two types of actions: *domain actions* that change the world (e.g., an action of downloading a paper from a url), and *sensing actions* that do not change the world but only return information about it (e.g., querying a bibliography). Each domain action has a list of *preconditions*, $\mathcal{P}$, and an encoding of the effects of the action as a set of literals, called the *assert list*, $\mathcal{A}$. Action preconditions identify the propositions necessary for executing the action. The propositions in $\mathcal{P}$ can include literals and quantified ψ-forms, where the term *quantified* is used informally to denote a ψ-form that uses at least one variable, and thus represents infinite number of ground instances. We assume that an action is deterministic and can change the truth-value of only a finite number of atoms, thus assert list contains literals only, and no quantified ψ-forms.

Sensing actions also have preconditions. Effects of the sensing are given by its *knowledge list*, denoted $\mathcal{K}$. The propositions in $\mathcal{K}$ are KW-ψ-forms.

Download(?p, ?s, ?u)
$\mathcal{P}: Url(?u), Source(?s), Paper(?p),$
$HasPaper(?u, ?s, ?p)$
$\mathcal{A}: Got(?p)$

FindMatchesInTitles(?b, ?kwd)
$\mathcal{P}: PrefBib(?b)$
$\mathcal{K}$: $[KW(\neg TitleUses(p, ?kwd) \vee \neg InCollection(p, ?b))]$

Figure 1: Example of Writer's Aid's domain and sensing actions. The variable p is implicitly universally quantified. Other variables are action schema parameters

Figure 1 provides examples of two Writer's Aid actions. *Download(?p, ?s, ?u)* is an action of downloading paper $?p$ from url $?u$ of source $?s$. Its preconditions are atoms, and its assert list consists of a single atomic effect that the paper is locally available. Another depicted action, *FindMatchesInTitles(?b, ?kwd)* is a sensing action that identifies all papers that according to bibliography $?b$ contain keyword $?kwd$ in their title. The precondition requires that $?b$ be a preferred bibliography. The effect of this action is encoded in the knowledge list that contains a quantified ψ-form, and states that as a result of this action the set of all papers in collection of bibliography $?b$ that use keyword $?kwd$ in the title will be known.

Conclusions

Representing a personalization task via a set of information goals addresses the problems with the way personalization is approached in most modern systems that are outlined in the beginning of this paper in the following ways:

- It leads to preference elicitation that occurs *within the context* of the particular task that requires personal information, thus informing the user of their choices at the time when the choices matter, motivating the response and ensuring its accuracy.
- Personalization occurs gradually at the times when the personal information is critical to the satisfaction of a user's goal and is initiated by the system, thus relieving the user from potentially time consuming task of specifying all preferences at once or searching for a hard to find option setting.
- Defining personalization task declaratively via information goals separates customization of the interface from the overall system architecture, making the interface more easily adjustable and extendable.

References

Babaian, T.; Grosz, B. J.; and Shieber, S. M. 2002. A writer's collaborative assistant. In *Proceedings of Intelligent User Interfaces'02*, 7–14.

Babaian, T. 2000. *Knowledge Representation and Open World Planning Using ψ-forms.* Ph.D. Dissertation, Tufts University.

Etzioni, O.; Golden, K.; and Weld, D. 1997. Sound and efficient closed-world reasoning for planning. *Artificial Intelligence* 89(1–2):113–148.

Manber, U.; Patel, A.; and Robison, J. 2000. The business of personalization: experience with personalization of Yahoo! *Communications of the ACM* 43(8):35–39.

Pednault, E. P. D. 2000. Representation is everything. *Communications of the ACM* 43(8):80–83.