Constraint Satisfaction Methods for Information Personalization

Syed Sibte Raza Abidi
Faculty of Computer Science, Dalhousie University, Halifax B3H 1W5, Canada
sraza@cs.dal.ca

Yong Han Chong
School of Computer Sciences, Universiti Sains Malaysia, Penang 11800, Malaysia
thestep@cs.usm.my

Abstract. Constraints formalize the dependencies in a physical world in terms of a logical relation among several unknowns. Constraint satisfaction methods allow efficient navigation of large search spaces to find an optimal solution that satisfies given constraints. This paper explores the application of constraint satisfaction methods to personalize generic information content with respect to a user-model. We present a constraint satisfaction based information personalization framework that (a) generates personalized information via the dynamic selection and synthesis of multiple information-snippets; and (b) ensures that the dynamically adapted personalized information is factually consistent. We present four constraint satisfaction methods that cumulatively work to maximize collaboration and minimize conflicts between a set of information-snippets in order to dynamically generate personalized information.

1 Introduction

Constraints arise in most areas of human endeavor and we are used to solving them in an unambiguous and efficient manner. Computationally, constraint satisfaction methods allow the efficient navigation of large search spaces to find an optimal solution that entails the assignment of values to problem variables subject to given constraints [1,2]. Constraint satisfaction programming has been successfully applied to many problem areas that demand the hard search for a solution, such as *configuration* [3], *planning* [4], *resource allocation* [5] and *scheduling* [6], and lately many new and interesting applications of constraint satisfaction are emerging.

The profusion of web-based information resources hosting large volumes of diverse information content offers a mixed outlook to users. On the one hand, there is comfort in the fact that information is available for use if and when needed, yet on the other hand there is an apprehension considering the effort required to sift and process the available information in order to achieve a meaningful impact. *Information Personalization* (IP) research attempts to alleviate the cognitive overload experienced by users in processing and consuming generic, non-focused information content [7]. Put simply, IP involves the dynamic adaptation of generic information content to generate personalized information content that is intelligently designed to suit an individual's demographics, knowledge, skills, capabilities, interests, preferences, needs, goals, plans and/or usage behavior [8, 9]. To date, there are a number of web-mediated information services that provide personalized information for a variety of reasons, including healthcare [10], customer relationships [11], product promotions, education [12] and tourism. At the forefront of such IP initiatives are adaptive hypermedia systems [13] that manifest a hybrid of artificial intelligence methods—in particular natural language processing, case-based [14], model-based, and rule-based methods—to provide a variety of IP methods and perspectives [15].

In our work we investigate the modeling of IP as a constraint satisfaction problem. In our view, IP is achieved by selecting multiple highly-focused information-objects, where each information-object may correspond to some aspect of the user-model, and appending these user-specific information-objects to realize a seamless personalized information package. The process of IP, therefore, can be modeled as a constraint satisfaction problem that involves the satisfaction of two constraints: (1) given a large set of available information-objects, the constraint is to select only those information-objects that correspond to the user-model; and (b) given the selection of multiple user-compatible information-objects, the constraint is to retain only those information-objects that cumulatively present a factually consistent view—i.e. the contents of the retained information-items do not contradict each other.

In this paper, we present an intelligent constraint-based *information personalization framework* that (a) generates personalized information via the dynamic selection of multiple topic-specific information-objects deemed relevant to a user-model [8]; and (b) ensures that the dynamically adapted personalized information, comprising multiple topic-specific information-objects, is factually consistent. We present a unique hybrid of adaptive hypermedia and variations of existing constraint satisfaction methods that cumulatively work to maximize collaboration and minimize the conflicts between a set of information-objects to generate personalized information.

2 The Problem of Information Personalization

From an adaptive hypermedia perspective IP is achieved at three levels: (i) *Content adaptation* involves both linguistic changes to the information content and changes to the composition of text fragments that jointly make-up the finished personalized hypermedia document; (ii) *Structure adaptation* involves dynamic changes to the link structure between the hypermedia documents; and (iii) *Presentation adaptation* involves changes to the physical layout of content within the hypermedia document [9].

Content adaptation is the most interesting and challenging strategy for IP, because it involves the dynamic selection of multiple information-objects that correspond to a given user-model, and then their synthesis using a pre-defined document template to realize a personalized information. We argue that although existing IP methods generate highly focused personalized information vis-à-vis the user-model, they do not take into account the possibility that the ad hoc synthesis of heterogeneous information-objects (*albeit* the information-objects are relevant to the user) might unknowingly compromise the overall factual consistency of the personalized information content.

Combining two information-objects can inadvertently lead to the generation of factually inconsistent information—i.e. one information-object stating a certain fact/recommendation whilst another information-object simultaneously contradicting the same fact/recommendation. We believe that in the absence of a content consistency checking mechanism, when multiple information-objects are synthesized, doubts may remain over the factual consistency of the personalized information.

Our definition of an IP problem therefore states that the scope of IP should not be limited to satisfying the user profile only, rather the IP strategy should also ensure that the personalized information content is factually consistent—i.e. no aspect of the personalized information content should be in contradiction with any other information simultaneously presented to the user. Hence, IP can be viewed as the satisfaction of two different constraints: (1) matching user-model attributes with information-object attributes to select user-specific information content; and (b) establishing information content consistency between multiple information-objects to ensure the factual consistency of the personalized information content.

2.1. Problem Specification

We approach the problem of IP at the content adaptation level. Our work is based on text fragment variants [11, 8], whereby a set of text fragments (or documents) are dynamically selected in accordance with the various aspects of a user profile. At runtime, the set of selected text fragments are systematically amalgamated to realize a hypermedia document containing personalized information. The problem of IP, from an optimization perspective, can therefore be specified as:

Given: (1) a *user-model* that comprises a number of user-defining attributes that describe the individual characteristics of a user; (2) a corpus of hypermedia documents called *Information Snippets* (IS). As the name suggests, each IS contains a text fragment of highly focused information that is pertinent to users with specific user-attributes. The IS are organized in a taxonomy that has four levels, as shown in Fig. 1.

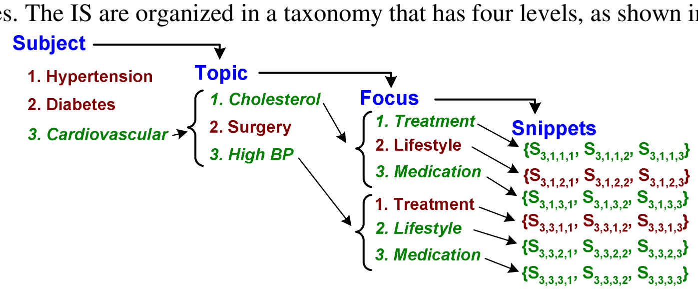

Fig. 1. A taxonomy of information snippets. A traversal through the taxonomy is shown by following the italicized text from subject to topic to focus to snippets.

For an exemplar healthcare IP problem, at the highest level the *Subject* can be broadly classified into cardiovascular disease, diabetes, hypertension, etc. Each subject is further classified into *Topics*, for instance cardiovascular disease can be described in terms of cholesterol management, heart surgery, diagnostics, high BP etc.

Each topic then entails multiple *Focus* areas, each focus area referring to a different aspect of the topic, for instance the different focus areas for cholesterol management are lifestyle, diet and medications. Finally, for each focus area there is a set of *Information Snippets*, where each IS contains information relevant to a specific focus area and targets specific user-attribute values such as age, gender, education level, etc.

Required: IP requires the automatic generation of the most *comprehensive*, *factually consistent* and *personalized* information package comprising a number of relevant IS that are systematically selected from the corpus and organized to yield a final personalized *Information Package.*

Constraints: The above three requirements translate into the following constraints: *Personalized*—the final information package should comprise all ISs that are consistent with the user-model; *Factual Consistency*—maintaining the personalized constraint, the final information package should ensure inter-IS consistency such that any two (or more) ISs should not give conflicting or inconsistent information; *Comprehensiveness*—maintaining the factual consistency constraint, the final information package should include the largest possible set of ISs that satisfy all constraints, and most importantly ensure that each focus area is minimally covered by a single IS.

Solution: The above problem specification brings to relief an interesting optimization problem, whereby the problem space on the one hand encompasses a wide diversity of users, whilst on the other hand a large volume of generic information content (in terms of ISs). The IP solution therefore involves searching the available ISs with respect to the user's characteristics, and selecting the largest possible set of relevant IS that jointly present a factually consistent view of the topic in question.

2.2. Operational Considerations

User-Model: A user-model comprises a set of user-defining attributes, each describing a particular characteristic of a user. Each user-attribute (UA) is represented as the tuple shown below:

$$\text{UA}(\textit{attribute, value, weight}) \quad 0 \leq \textit{weight} \leq 1,\ 0 \rightarrow \text{absent},\ 1 \rightarrow \text{present}$$

Where *attribute* refers to a user characteristics such as age, gender; *value* denotes the numeric or symbolic measurement of the attribute; and *weight* refers to the presence or absence of that particular attribute's value in the user-model. For example, UA(age, 40, 1) implies that the age of the user equaling 40 is valid. And, UA(allergy, pollen, 0) implies that the user does not have allergy to pollen.

Information Snippet (IS): An IS is represented in the form of a *conditional frame* that involves the binding of information content with a set of conditions [16]. Each IS is composed of two sections: (a) *Content section* that withholds the information content; and (b) *Condition section* that specifies the conditions for the selection of the document. The condition section comprises two types of conditions: (a) *Snippet-Selection Conditions* (SSC) that are compared with the user's model in order to determine whether the said IS is relevant to the user. An IS is selected if *all* SSC are satisfied; and (b) *Snippet-Compatibility Conditions (SCC)* determine whether the said IS can mutually co-exist with other selected IS. An IS is selected if *all* SCC are satisfied. Both these conditions are representation by the tuple:

SSC/SCC (*context, value, weight*)

$0 \le weight \le 1$, $0 \rightarrow$ not recommended, $1 \rightarrow$ recommended

In the condition tuple, the *context* determines the nature of the condition, *value* states the text or numeric description of the condition, and *weight* defines the degree of the condition ranging from 0 to 1. For example SSC(allergy, pollen, 0) means the context of the condition pertains to allergies, the specific value of the context is pollen, and the weight being 0 implies not recommended. Hence, an IS with the above SSC cannot be selected for a user who has an allergy to pollen. Similarly, the SCC(drug, aspirin, 0) means the IS is compatible with all IS that *do not* recommend the drug named aspirin.

3 Modeling Information Personalization as a Constraint Satisfaction Problem

3.1. Constraint Satisfaction: An Overview

Mathematically speaking, constraints formalize the dependencies in a physical world in terms of a logical relation among several unknowns (or variables), each taking a value from a defined domain. In principle, a constraint restricts the possible values that the variables can take whilst solving a problem. *Constraint programming* solves problems by stating constraints about the problem area and consequently finding solutions that may 'satisfy' all the constraints. A *Constraint Satisfaction Problem* is defined by a tuple $P = (X, D, C)$ where $X=\{X_1, \ldots , X_n\}$ is a finite set of *variables*, each associated with a domain of discrete values $D = \{D_1, \ldots, D_n\}$, and a set of constraints $C = \{C_1, \ldots, C_l\}$. Each constraint C_i is expressed by a relation R_i on some subset of variables. This subset of variables is called the *connection* of the constraint and denoted by $con(C_i)$. The relation R_i over the connection of a constraint C_i is defined by $R_i \subseteq D_{i1} \times \ldots \times D_{ik}$ and denotes the tuples that satisfy C_i. A solution to a constraint satisfaction problem is an assignment of a value from its domain to every variable, in such a way that every constraint is satisfied [1, 2, 3]. This may involve finding (a) just one solution with no preferences, (b) all solutions, or (c) an optimal solution given some objective function defined in terms of some or all of the variables.

Solutions to a constraint satisfaction problem can be found by systematically searching through the possible assignments of values to variables using several different approaches. Popular approaches include the *Generate-and-Test* methods [17] that systematically generate each possible value assignment and then test to see if it satisfies all the constraints, and *Backtracking* methods [18] that incrementally attempt to extend a partial solution toward a complete solution. Both search methods guarantee a solution, if one exists, or else prove that the problem is insoluble [19].

Generate-and-Test methods generate all the possible solutions in the search space and then test each solution to determine whether it is the right solution. In doing so, each possible combination of the variable assignments is systematically generated and tested to see if it satisfies all the constraints. The first combination that satisfies all the constraints is taken as the solution. Backtracking search methods sequentially instantiate the variables in some order, and as soon as all the variables relevant to a constraint are instantiated, the validity of the constraint is checked. If the constraint is not satisfied, backtracking is performed to the most recently instantiated variable that still has alternative values available for examination. In this way, backtracking has the advantage of extending a partial solution that specifies consistent values for some of the variables towards a search for a complete solution [17, 18, 19]. Another approach for constraint satisfaction involves *Consistency* techniques that detect inconsistent values that cannot lead to a solution, and thus prune them from the search space to make the search more efficient [20, 21]. *Node Consistency* is the simplest consistency technique that works as follows: The node representing a variable V in a constraint graph is *node consistent* if for every value X in the current domain of V, each unary constraint on V is satisfied. If the domain D of a variable V contains a value Z that does not satisfy the unary constraint on V, then the instantiation of V to Z will always result in failure. This implies that node inconsistency can be eliminated by simply removing those values from the domain D of each variable V that do not satisfy the constraint on V.

3.2. Our CS-Based Information Personalization Approach

Given a subject and its constituent topics, we provide information personalization at the topic-level. For each topic in question, the search strategy is to select the most relevant and consistent IS for all its focus areas (see taxonomy shown in Fig. 1).

We define IP in a constraint satisfaction context as (a) a set of focus areas for a given topic, represented in terms of *focus-variables* $X=\{x_1,...,x_n\}$, where for each focus-variable x_i, there is a finite set of (focus-specific) IS. The set of IS associated with each focus-variable is deemed as its domain, D_i; (b) a user-model represented as a single-valued *user-variable*; and (c) and two types of constraints—*user-model constraint* and *co-existence constraint*. A solution to our constraint satisfaction problem is the systematic selection of the largest subset of IS associated with each topic—this is achieved by selecting the largest subset of IS for each focus-variable associated with the said topic—in such a way that the given user-model and co-existence constraints (amongst all selected IS) are fully satisfied. Such a constraint satisfaction solution can be obtained by searching the domain for each focus-variable. Our constraint satisfaction approach for searching the solution is given as follows:

Step 1-Selection of user-specific information content: The user-model attributes forms the basis for selecting user-specific IS. Node-consistency based techniques are used to solve the *user-model constraint* by satisfying the snippet-selection conditions of each IS (where the IS is related to the given topic by a focus variable) with the user-attributes noted in the user-model. We collect a *candidate-IS set* that comprises all possible (topic-specific) ISs that are relevant to the user-model (shown in Fig. 2b).

Step 2-Selection of 'Core' information content: Given the candidate-IS set, it is important to ensure that the selected ISs can potentially co-exist with each other without causing any factual inconsistency. Hence the next step is to establish the minimum information coverage that is factually consistent—i.e. establishing the *core-IS set* which includes a single IS for each focus area in question. We use backtracking search to satisfy the *co-existence constraints* by globally satisfying the snippet-compatibility conditions for all the IS in the candidate-IS set. Any IS that is deemed factually inconsistent with the rest of the IS is discarded. The resulting *core-IS set* (as illustrated in Fig. 2c) depicts the minimum coverage of factually consistent information whilst also satisfying the requirement for *comprehensiveness*—i.e. to minimally cover each focus area with a single IS for all topics in question.

The rationale for generating a core-IS set is to initially establish a baseline of factually-consistent ISs that meet the comprehensiveness requirement. The core-IS set provides limited information coverage, but more importantly the information is factually consistent—our thinking being that it is better to give less information but ensure that it is consistent, than to give more information that maybe potentially inconsistent. Having established a baseline (or minimum) factually consistent information, in the next steps we attempt to build on the core-IS set to extend the information coverage.

Step 3-Selection of 'Extended' information content: Given the core-IS set, we next attempt to maximize its information coverage by including previously non-selected candidate-ISs (in step 2) to the core-IS set, whilst ensuring that the overall factual consistency is maintained. We use the stochastic generate-and-test method to 'stochastically' search for previously non-selected candidate-ISs that satisfy the co-existence constraint with the core-IS set. If the co-existence constraint is satisfied, the candidate-IS is included to the core-IS set resulting in an *extended-core-IS set* which will then be used as the baseline for future inclusions of other candidate-ISs. Note that if no additional candidate-IS can be included to the core-IS set then the extended-core-IS set equals the core-IS set. The outcome of this step is a more optimal extended-core-IS set that represents the new, yet potentially larger than before, minimum information coverage that satisfies both the user-model and co-existence constraints (shown in Fig. 2d). In the next step we attempt to maximize the information coverage.

Step 4-Selection of 'Optimal' information content: The generation of the core-IS set and the follow-up extended-core-IS set involved the use of stochastic search algorithms that were solely designed to satisfy the co-existence constraints between the candidate-IS, without checking the possibility that the selected candidate-IS may in turn block the future inclusion of other candidate-IS to the core- and extended-core-IS sets. It is fair to assume that due to the stochastic nature of the solution, there may exist the possibility that a particular candidate-IS may satisfy the prevailing co-existence constraint situation at that time and become a member of the core- or extended-core-IS set, but being inconsistent with a large number of non-selected candidate-ISs it may block their potential inclusion to the extended-core-IS set, thus contributing to a sub-optimal solution. Having said that, the exclusion of a single sub-optimal candidate-IS from the extended-core-IS set may enable the potential inclusion of multiple non-selected candidate-ISs to the extended-core-IS set, whilst still maintaining the co-existence constraints and the comprehensiveness requirement.

In order to further optimize the information coverage, our approach is to explore the possibility of replacing a single sub-optimal IS in the extended-core-IS set with multiple non-selected candidate-IS. This is achieved by our novel information optimization mechanism, termed as *snippet swapping*. The snippet swapping mechanism generates the most optimal information coverage in terms of the final *presentation-IS set* (shown in Fig. 2e), that (a) maintains the co-existence constraints, and (c) ensures that each focus area (for all selected topics) is represented by at least one IS. Note that if snippet swapping is not possible then the presentation-IS set equals the extended-core-IS set. In conclusion, the optimized presentation-IS set is the final CSP solution.

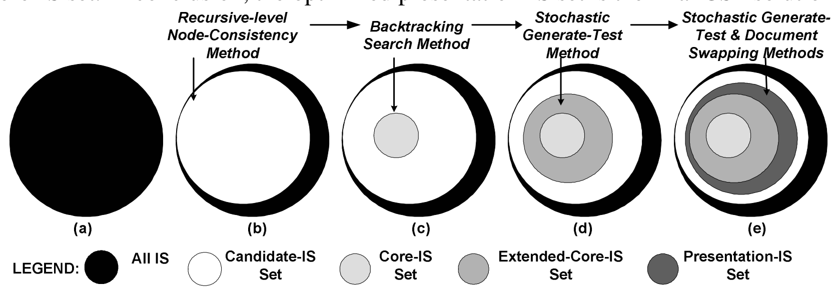

Fig 2: Schematic representation of the different stages of the CSP solution, highlighting the respective maximization of the information coverage at each progressive stage.

4 Constraint Satisfaction Methods for Information Personalization

In line with the abovementioned IP approach we have developed variants of consistency-checking techniques and search algorithms to generate the personalized presentation-IS set. In the forthcoming discussion we present our variants of constraint satisfaction methods that are used to solve the user-model constraint to generate the candidate-IS set, and the co-existence constraints to generate the core- and extended-core IS sets, and the snippet-swapping method to generate the presentation-IS set.

4.1 User-model constraint satisfaction: Generating the candidate-IS set

A user-model constraint between a *focus-variable* and a *user-variable* is satisfied when all the IS in the domain of the focus-variable are consistent with the user-model. The general idea is to compare the snippet-selection conditions (SSC) for each IS with the user-attributes (UA) listed in the user-model (UM) as follows, $(context, value)_{SSC}^{IS} = (attribute, value)_{UA}^{UM}$. We calculate a *conflict value* (CV), as shown below, between the SSC and UA to determine constraint satisfaction. A low CV value implies that the user-model constraint has been satisfied and that the IS is deemed relevant to the user, whereas a high CV value denotes the irrelevance of the IS to the user. The acceptance level of CV is a parameter that can be set the user to determine the desired severity of the SSC. The CV is the modulus of the difference between the weights of the SSC and the matching UA, and is calculated as follows:

$$CV_{SSC}^{UA} = \left|(weight)_{SSC}^{IS} - (weight)_{SSC}^{UM}\right|, \ (context, value)_{SSC}^{IS} = (attribute, value)_{UA}^{UM}$$

$$0 \le CV_{SSC}^{UA} \le 1 \ ; \ where\, 0 \rightarrow const.\ satisfied\, ,\ 1 \rightarrow const. not\ satisfied$$

To satisfy the user-model constraint we employ a variation of CSP node-consistency technique—the *recursive-level node-consistency algorithm* [2]. The working of our modified recursive-level node-consistency algorithm is as follows: for each focus-variable, if the domain contains an IS that is inconsistent towards the user-model, then that particular IS is removed from the domain. Eventually, only those IS that are consistent with the user-model are retained in each focus-variable's domain and the resulting set of user-specific IS are regarded as the *candidate-IS set*.

Algorithm Recursive-level Node Consistency

```
for focus-var1 to focus-varm{m = number of focus areas}
   for IS1 to ISn {n = no. of IS in the domain of focus-vari}
      test UMC  {UMC = user model constraint}
      if UMC not satisfied {inconsistent with user-model}
             discard ISi
      endif
   endfor
endfor
```

4.2. Co-existence constraint satisfaction I: Generating the core-IS set

Co-existence constraints between two focus-variables need to be satisfied to ensure that their respective selected ISs are factually consistent with each other. In practice, co-existence constraints between two focus-variables$_{A\&B}$ are satisfied if the selected ISs from the domain of focus-variable$_A$ are consistent with the selected ISs from the domain of focus-variable$_B$. Two SCC are only comparable if they both have the same content and value, as follows: $(context, value)_{SCC_A}^{IS_A} = (context, value)_{SCC_B}^{IS_B}$. The SCC of an IS is satisfied with respect to the SCC of another IS. A co-existence constraint is not-satisfied when the conflict value (CV) exceeds a predefined user threshold.

$$CV_{SCC_A}^{SCC_B} = \left| (weight)_{SCC_A}^{IS_A} - (weight)_{SCC_B}^{IS_B} \right|, (context, value)_{SCC_A}^{IS_A} = (context, value)_{SCC_B}^{IS_B}$$

$$0 \le CV_{SCC_A}^{SCC_B} \le 1 \text{ ; } where\, 0 \rightarrow const.\ satisfied\ ,\ 1 \rightarrow const.not\ satisfied$$

To satisfy co-existence constraints leading to the generation of the core-IS set we employ a Backtracking (BT) search method. The BT method searches the candidate-IS space to generate the core-IS set by (i) choosing an un-instantiated focus-variable, i.e. no IS has yet been assigned to the focus-variable; (ii) choosing a candidate-IS from the domain of the un-instantiated focus-variable; (iii) checking whether the candidate-IS is consistent with ISs that have already been selected to instantiate the other focus-variables; (iv) if the candidate-IS is consistent—implying that the co-existence constraint is satisfied—it is selected by instantiating the focus-variable, else the next candidate-IS within the domain of the same focus-variable is examined. Given that the co-existence constraint cannot be satisfied because all the candidate documents for a focus-variable have been checked, backtracking is performed to select the most recently instantiated focus-variable that may still have some alternative candidate-ISs and then search forward again based on the new instantiation of the said focus-variable. Successful BT search ensures that each focus-variable is instantiated with an IS, thus satisfying the minimum comprehensiveness requirement, and resulting in the *core-IS set*. The order in which the topics are searched can be based on the following schemes: (1) Original chronological order of the topics; (2) Randomly selecting the next topic to search.; (3) User-specified search order of the topics; important topics are search first followed by the less significant topics; (4) Partial user-specified order (the starting topic and maybe a few others are given) and the remaining topics are selected in a random order.

4.3. Co-existence constraint satisfaction II: Generating the extended-core-IS set

Extension of the core-IS set to the potentially larger extended-core-IS set is performed via the *Stochastic Generate and Test* (S-GT) method. The motivation for generating the extended-core-IS set is to maximize the current information coverage by selecting previously non-selected candidate-IS that do not violate the a priori established factual consistency of the core-IS set.

The working of the S-GT method is as follows: the non-selected candidate-IS are randomly sequenced in N different groups. Each group of ISs is then systematically searched based on the sequence of the constituent ISs in the group in an attempt to include more candidate-IS into the core-IS set without violating the co-existence constraint. Consequently, N extended-core-IS sets are generated, whereby the extended-core-IS set with the most ISs is selected. We argue that the S-GT method is suitable for this purpose because of its stochastic nature in selecting focus-variables and evaluating the ISs within their domain in a manner that avoids the 'unfair' effects resulting from a sequenced evaluation of ISs as practised by most search algorithms.

4.4. Snippet Swapping: Generating the presentation-IS set

The information coverage of the extended-core-IS set can be further increased by including more non-selected candidate-IC, but at this stage this is only possible by removing an IS in the extended-core-IS set. The basic idea is to 'swap' a single IS in the extended-core-IS set with multiple candidate-ISs—this is reflective of the situation when a single IS in the extended-core-IS set is factually inconsistent with multiple candidate-IS, hence it is single-handedly blocking the inclusion of multiple candidate-ISs to the extended-core-IS set. The snippet swapping algorithm, given below, explains the thinking behind the snippet swapping mechanism.

Algorithm Snippet Swapping

```
for each IS_A in the extended-core-IS set
   identify the non-selected candidate-ISs that are
   inconsistent to IS_A
   if size of non-selected candidate-ISs N > 1
      if IS_A is not the only IS selected for a focus-variable
        apply S-GT algorithm to the non-selected candi-
        date-ISs to generate N sets
        if size of the largest set of candidate-IS C > 1
             discard IS_A
             append C to the extended-core-IS set
        endif
      endif
   endif
 endfor
```

The snippet swapping mechanism extends the information coverage whilst still maintaining the factual consistency and comprehensiveness requirements of the result presentation-IS set.

5 Generating Personalized Healthcare Information

We present a working example of constraint satisfaction based IP as per our approach discussed earlier. The scenario involves a person suffering from two health problems—i.e. high BP and arthritis—and we need to provide personalized healthcare information based on his user-model given in Table 1.

Table 1. An exemplar user-model

Health Problemss					
1. High Blood Pressure		2. Arthritis			
User Attributes (UA)					
Attribute	**Value**	**Weight**	**Attribute**	**Value**	**Weight**
Age	45	1	Medication	DrugY	1
Gender	Male	1	Lifestyle	Smoker	1
Education	Graduate	1	Lifestyle	Active	0
Family History	Diabetes	0	Allergy	Pets	1
Medication	DrugX	0	Allergy	Pollen	0

As per our IS organization taxonomy (given in Fig. 1), the two topics are *high BP* and *arthritis*, each having two focus areas namely *treatment* and *medication*. Table 2 illustrates the set of IS available for each focus area for each topic. We need to define the focus-variable (*focus_var*) representing each focus area, such that the domain for each focus-variable comprises the ISs that correspond to the focus. Due to space limitations we will not be able to show the processing for each focus variable, however for illustration purposes the outcome of the CS methods for focus_var1 are shown.

Step 1- Generate Candidate-IS set: This involves the satisfaction of user-model constraints using the node-consistency algorithm. Table 3 shows the candidate-IS set for focus_var1, whereby only the ISs that are relevant to the user-model are selected.

Step 2- Generate Core-IS set: This step involves the satisfaction of the co-existence constraints for each IS (not be shown due to lack of space). Table 4 shows the core-IS set derived from the candidate-IS set for each focus variable. Note that the core-IS set comprises the first ISs in the focus_var list —i.e. HT1, HM1 and AM1—for the focus_var1 (HT), focus_var2 (HM) and focus_var4 (AM). This is because the search algorithm starts with the first IS in the focus_var list. Interestingly enough, for the focus_var3 the third IS—i.e. AT3—is selected because firstly AT1 was in conflict with HM1 and then secondly AT2 was in conflict with HT1. Since, both HM1 and HT1 were already a member of the core-IS set when the evaluation for AM was concluded, hence an IS was chosen that could co-exist with the a priori members of the developing core-IS set. The affect of sequencing of IS for evaluation and subsequent selection is addressed in the snippet-swapping stage.

Table 2. IS for the topics high blood pressure and arthritis. Also shown is the defintion of variables (focus_var) for each focus area in the realm of a topic.

Topic	Focus	IS	Variable ::{Domain}
High Blood Pressure (H)	Treatment (T)	HT1, HT2, HT3, HT4	focus_var1::{HT1, HT2, HT3, HT4}
	Medication (M)	HM1, HM2, HM3, HM4	focus_var2::{HM1, HM2, HM3, HM4}
Arthritis (A)	Treatment (T)	AT1, AT2, AT3, AT4	focus_var3::{AT1, AT2, AT3, AT4}
	Medication (M)	AM1, AM2, AM3, AM4	focus_var4::{AM1, AM2, AM3, AM4}

Table 3. The candidate-IS set for the topic high blood pressure and focus area is treatment.

Doc	Snippet Selection Condition	Matching User Attribute	CV	Status
HT1	<gender, male, 1>	<gender, male, 1>	0	**Retained**
HT2	<family history, diabetes, 1>	<family history, diabetes, 0>	1	Discarded
HT3	<medication, DrugX, 1>	< medication, DrugX, 0>	0	**Retained**
HT4	< allergy, seafood, 1>	<condition, pregnant, 0>	1	Discarded

Step 3 – Generate the Extended-Core-IS set: Next, we attempt to increase the information coverage of the core-IS set by applying the stochastic generate and test algorithm as per our approach for generating the extended-core-IS set. Table 5 shows the three random sets of non-selected candidate-IS, whereby the third random set is shown to best maximize the information coverage.

Note the stochastic nature of the search as the random ordering of the ISs for SCC satisfaction affects the outcome. In the third set, AT4 which is inconsistent with both HM4 and HT3 was positioned after HM4 in the random set. This enabled HM4 to be selected first instead of AT4 and thus blocked AT4 to be selected subsequently. Without AT4 in the extended-core-IS set it was possible for HT3 to be next selected. Note that this situation was not possible for the first two random sets. AM2 was not selected because it was in conflict with HM1 which was a member of the core-IS set. AT1 and AT2 are still non-selectable as they conflict with two members of the core-IS set.

Step 4- Generate the Presentation-IS set: Finally, we attempt to achieve optimal information coverage by applying the snippet swapping mechanism to generate the presentation-IS set. Table 6 shows the extended-core-IS set (comprising 8 ISs) together with their conflicts with IS discarded during BT and S-GT search. HM1 has been detected to be blocking two candidate-ISs—i.e. AT1 and AM2. Since the *Topic:Focus* area for High *BP:Medication* is represented by both HM3 and HM4 in the extended-core-IS set, it is possible to swap HM1 with AT1 and AM2 without disturbing the factual consistency and still maintaining the completeness requirement. As a result we get an optimal presentation-IS set (as shown in Table 7) that is larger than the initial extended-core-IS set. The resultant presentation-IS set is the solution of the IP problem and represents the personalized and factually consistent information suited for a specific user-model.

Table 4. Given the candidate-IS set (coverig all focus areas), we illustrate the core-IS set derived using backtracking search method. Also shown are the non-selected candidate-IS.

Topic-variables	Domain (Candidate-IS set)	Core-IS set	Non-selected Cand.-IS
focus_var1	HT1, HT3	**HT1**	HT3
focus_var2	HM1, HM3, HM4	**HM1**	HM3, HM4
focus_var3	AT1, AT2, AT3, AT4	**AT3**	AT1, AT2, AT4
focus_var4	AM1, AM2, AM3	**AM1**	AM2, AM3

Table 5. An extended-core-IS set resulting from S-GT search over three random sets of IS

Non-selected candidate-IS arranged in a random order	IS selected	IS discarded	Size
AT4, HM3, AM3, AT1, HM4, AM2, HT3, AT2	AT4, HM3, HT3	AM3, AT1, HM4, AM2, AT2	3
AM2, AT1, AT4, HM4, HT3, AT2, HM3, AM3	AT4, HT3, HM3	AM2, AT1, HM4, AT2, AM3	3
HM4, AT4,AT2, HT3, HM3, AM2, AT1, AM3	**HM4, HT3, HM3, AM3**	**AT4, AT2, AM2, AT1**	**4**

Table 6. Extended-core-IS set *before* optimization. The italized IS are members of the core-IS set, whereas the others were added later during the extended-core-IS generation step.

		Non-selected candidate-ISs				
		AT1	**AT2**	**AT4**	**AM2**	**# of Conflicts**
Presentation Set (size =8)	***HT1***	-	X	-	-	**1**
	HT3	-	-	X	-	**1**
	HM1	X	-	-	X	**2**
	HM3	-	-	-	-	**0**
	HM4	-	-	X	-	**1**
	AM1	-	-	-	-	**0**
	AM3	-	-	-	-	**0**
	AT3	-	-	-	-	**0**
Conflicts		**1**	**1**	**2**	**1**	

Table 7. An optimal presentation-IS set *after* optimization.

		Non-selected ISs			**# of Conflicts**
		AT2	**AT4**	**HM1**	
Presentation Set (Size = 9)	***HT1***	X	-	-	**1**
	HT3	-	X	-	**1**
	HM3	-	-	-	**0**
	HM4	-	X	-	**1**
	AM1	-	-	-	**0**
	AM3	-	-	-	**0**
	AT3	-	-	-	**0**
	AT1	-	-	X	**1**
	AM2	-	-	X	**1**
Conflicts		**1**	**2**	**2**	

5.1 Evaluation

The evaluation of the featured IP method focused on establishing the completeness and factual consistency of the information package. The computational complexity of the search methods were not measured as it was not deemed to be the most pressing issue at this stage, however we will present the computational complexity of the various methods in a separate publication. We anticipated that the random nature of the CS search methods might have a significant bearing on the final output, because the manner in which the initial IS are selected determines the overall makeup of the final output. For that matter, the document swapping method introduced here provides an opportunity to re-visit the selected IS (extended core-IS set) and to optimize the presentation set. The experimental data comprised: 10 topics each with 2 focus areas; 70 IS each with constraints; and 10 controlled user-models. Given the experimental data, experiments were carried out to evaluate the completeness and consistency of the final output. Analysis of the output—i.e. the personalized information package—indicated that whenever the completeness criteria was satisfied all the IS present in the presentation set were found to be consistent with each other. This observation vindicates the efficacy of the CS methods deployed to achieve IP.

6 Concluding Remarks

Person-specific customization of information viz. a user-model is a complex task that necessitates a systematic, pragmatic and multifaceted strategy. In this paper we presented and demonstrated an IP framework that purports a unique hybrid of adaptive hypermedia and constraint satisfaction methods. We have demonstrated the successful application of constraint satisfaction methods for information personalization that offers an alternate and interesting perspective to research in both information personalization and application of constraint satisfaction methods.

In conclusion, we believe that this is the first step towards the incorporation of constraint satisfaction within an information personalization paradigm. Also, the realization to ensure factual consistency when amalgamating heterogeneous information will lead to interesting research in adaptive information delivery systems. Finally, we believe that the featured IP approach can be used for a variety of E-services for education material customization, stock market reporting and advice, tourist information and so on; the only limitation is the specification of co-existence constraints which demands human expert involvement—a likely bottleneck.

References

[1] Tsang E, *Foundations of constraint satisfaction*. Academic Press, London, UK. 1993.

[2] Barták R, Constraint programming: In pursuit of the holy grail. *Proceedings of the Week of Doctoral Students (WDS99),* Part IV, MatFyzPress, Prague, 1999, pp. 555-564.

[3] Sabin D, Freuder E, Configuration as composite constraint satisfaction. *Proceedings of the Artificial Intelligence and Manufacturing Research Planning Workshop*, 1996, pp.153-161.
[4] Stefik M, Planning with constraints (MOLGEN: Part 1). *Artificial Intelligence*, Vol. 16(2), 1981, pp. 111-140.
[5] Sathi A, Fox MS, Constraint-directed negotiation of resource allocations. In *L. Gasser & M. Huhns (Eds.) Distributed Artificial Intelligence Volume II,* Morgan Kaufmann, San Mateo, 1989, pp. 163-194.
[6] Fox M, *Constraint-Directed Search: A Case Study of Job-Shop Scheduling*. Morgan Kaufmann, London, 1987.
[7] Zirpins C, Weinreich H, Bartelt A, Lamersdorf W, Advanced concepts for next generation portals. *Proceedings of 12th International Workshop on Database and Expert Systems Applications*, 3-7 Sept. 2001, pp. 501 –506.
[8] Fink J, Kobsa A, Putting personalization into practice. *Communications of the ACM*, 2002, Vol. 45(5).
[9] Perkowitz, M, Etzioni O, Adaptive web sites. *Communications of the ACM*, 2000, Vol. 43(8), pp. 152-158.
[10] Abidi SSR, Chong Y, Abidi SR, Patient empowerment via 'pushed' delivery of personalized healthcare Educational content over the internet. *Proceedings of 10th World Congress on Medical Informatics*, 2001, London.
[11] Kobsa A, Personalized hypermedia presentation techniques for improving online customer relationships. *Knowledge Engineering Review,* Vol. 16(2), 1999, pp. 111-155.
[12] Henze N, Nejdl W, Extendible adaptive hypermedia courseware: integrating different courses and web material. In *P. Brusilovsky, O Stock & C Strappavara (Eds.) Adaptive Hypermedia and Adaptive Web-based Systems*, Springer Verlag, 2000, pp. 109-120.
[13] Brusilovsky P, Kobsa A, Vassileva J (Eds.), *Adaptive Hypertext and Hypermedia*. Kluwer Academic Publishers, Dordrecht, 1998b.
[14] Abidi SSR, Designing Adaptive Hypermedia for Internet Portals: A Personalization Strategy Featuring Case Base Reasoning With Compositional Adaptation. In FJ Garijo, JC Riquelme & M Toro (Eds.) *Lecture Notes in Artificial Intelligence 2527: Advances in Artificial Intelligence (IBERAMIA 2002)*. Springer Verlag: Berlin, 2002. pp. 60-69.
[15] Zhang Y, Im I, Recommender systems: A framework and research issues. *Proceedings of American Conference on Information Systems*, 2002.
[16] Boyle C, Encarnacion AO, MetaDoc: An adaptive hypertext reading system. *User Models and User Adapted Interaction,* Vol. 4(1), 1994, pp. 1-19.
[17] Dechter R, Pearl J, Network based heuristics for constraint satisfaction problems. *Search in Artificial Intelligence,* Springer Verlag, Berlin, 1988, pp. 370-425.
[18] Kumar V, Algorithms for constraint satisfaction problems: A survey. *AI Magazine*, 1992, Vol. 13(1), pp. 3-44.
[19] Grzegorz K, van Beek, P, A theoretical evaluation of selected backtracking algorithms. *Artificial Intelligence,* 1997, Vol. 89, pp. 365-387.
[20] Mackworth AK, Consistency in networks of relations. Artificial Intelligence, 1977, Vol. 8, pp. 99-118.
[21] Freuder E, A sufficient condition for backtrack-free search. *Communications of the ACM*, 1982, Vol. 29(1), pp. 24-32.