International Journal on Advanced Science Engineering Information Technology

Vol.7 (2017) No. 1
ISSN: 2088-5334

An Ontological Approach to Constraint-Based Design Case Adaptation

Norlia Yusof [#], Shahrul Azman Noah[#], Samirah Taufiq Wahid[*]

[#]*Knowledge Technology Group, Centre of Artificial Intelligence Technology, Faculty of Information Science and Technology,*
University of Kebangsaan Malaysia, 43600 Bangi, Malaysia
E-mail: norlia@siswa.ukm.edu.my; shahrul@ukm.edu.my

[*]*School of Health Care Sciences, Faculty of Health Sciences, University of Kebangsaan Malaysia,*
Jalan Raja Tun Muda Abdul Aziz, 50300 Kuala Lumpur, Malaysia
E-mail: samirah@fsk.ukm.my

Abstract— **Designing tasks in case-based reasoning requires the use of case adaptation due to its novelty characteristic. In this paper, constraint satisfaction is used to generate potential solutions for design case adaptation. An ontological approach is proposed to perform the inference process to satisfy the multiple design constraints. Domain application is the dietary menu planning for diabetics. Results show that the dietary menu planning designed by the proposed approach is better than the conventional approach. Both the physical and aesthetic constraints were satisfied by the proposed approach.**

Keywords— **design; case-based reasoning; Case Adaptation; constraint satisfaction; ontology; dietary menu planning**

I. Introduction

Case-based reasoning (CBR) is one of the problem-solving methods for design tasks. This approach is known as case-based design (CBD). Case adaptation is an important subtask for CBD due to the novelty element of each new design solution. We propose for the use of the semantic technology approach to carrying out the design case adaptation. In semantic technology, ontology is the prerequisite.

Substitution and structural transformation are two types of adaptation methods [1]. The first type replaces the value of an old solution, whereas the latter type modifies its structure using insertion or deletion operation. The substitution alone cannot support the design case adaptation task because if only the value of some solutions is changed, it does not lead to innovation of design. Thus, the novelty characteristic that lies in the design demands for a structural transformation adaptation. However, substitution can be incorporated with structural transformation transformations to support the design case adaptation framework [2].

In design, problem requirements are defined in terms of the constraints that must be satisfied. A constraint can also impose by the domain. This situation is referred to as hard constraints, or also known as physical constraints. The other type of constraints is soft constraints. It has softer restrictions compared to the first type, such as personal preferences. It is not compulsory to satisfy soft constraints, however, by doing so can yield better design solutions. The solutions are said to be optimum if all the hard and soft constraints have been satisfied. Thus, it is important for design task to achieve both of the constraints because the higher-quality designs are preferable to lower ones.

Case combination is one of the conventional approaches towards design case adaptation [3]. This approach was inspired by complex design problems where one existing case could not afford to solve the new design requirements. Hence, the combination features of partially matching cases might overcome this situation. This technique is guided using cases only, while case adaptation is a knowledge-intensive task, this contradiction indicates it needs another case adaptation technique to support the whole case adaptation process. For example, researcher [4] applied subcase combination and rule-based transformation for their case adaptation framework. Subcase combination is one of the techniques in case combination [5]. Another limitation in case combination is inconsistencies may happen when design values from several cases are combined into one design solution. Thus, it requires extra knowledge to check and repair the problematic merging values. Researcher [6] used minimum-conflict repair heuristic to any inconsistency values until they reach a feasible solution. On the other hand, researcher [7] used specialized heuristics to repair the inconsistency values of synthesised solutions. Due to the limitations, it is beneficial for a designer to implement the knowledge-intensive case adaptation techniques.

We have applied the proposed approach to a dietary menu planning for a person with diabetes. This domain reflects the design task where a multiple of constraints needs to be satisfied. Among these constraints are the physical constraints where the design task needs to fulfil the food groups allotted in the meal exchange table (MET) and to substitute a forbidden food, the preference constraints such as the cultural food customs, and the common sense constraints in the context of food accompaniment.

Our approach toward design case adaptation is knowledge-intensive. The knowledge is in the form of constraints. Reference [2] classifies the use of constraints for design case adaptation into two dimensions. The first dimension categorizes the constraints whether they are specific for each case or are general to a domain. The second dimensions categorize the constraints role whether to generate possible solutions to a new problem or evaluate the solutions. From these dimensions, four categories of constraint-based design case adaptation are derived. The first category is case-specific constraints where constraints used for solution generation. The projects of FAMING and CADRE [8] and COMPOSER[6] falls into this category. The second category is general domain constraints where constraints used for solution generation. Our work falls into this category where constraint are generally imposed by the domain, and it is used for generating the solutions. The work carried out by [9] and [10] are belong to this category. The third category is case-specific constraints where constraint used for solution evaluation. So far, there is no research implemented yet for this category. The last category is general domain constraints where constraints used for solution evaluation. GENCAD by [11] falls under this category.

An ontological approach is proposed to satisfy the constraints imposed by the problem requirements and the domain application. We want to take advantage of description logic (DL) reasoning mechanism to fulfil the constraint-based design case adaptation. The research done by [12] is by far the closest to our own work, where they applied the DL formalism i.e. subsumption, instantiation and concept satisfiability to the reformulation model for protocol adaptation. Prior to this work, researchers [13] have formalized one of the techniques in substitution method i.e. specialized search using ontology. The hybridization between CBR and ontology mark-up language or known as semantic Web were reviewed by [14].

The objective of this paper is to demonstrate how an ontological approach satisfies both the physical and aesthetic constraints using structural transformation and substitution adaptations. An aesthetic constraint consists of preference and common sense constraints. With this aim, the paper is organized as follows. Section 2 explains the steps taken to implement the proposed approach and demonstrates the inference process to satisfy the multiple design constraints. Section 3 reveals the results gained from the experiment. Finally, Section 4 concludes the work of the proposed approach.

II. Material and Algorithm

The first step taken in this study was the modelling of the domain knowledge of the Malaysian food composition using ontology. The food composition ontology was developed based on the OD101 methods [15]. Protégé 4.3 (http://protege.stanford.edu/) was used as the ontology editor environment. The details of the food composition ontology modelling were discussed in [16] and [17].

The second step was the development of the CBR subtasks, which are case representation, retrieval, and adaptation. The CBR shell was chosen to implement the CBR engine as it offers rapid prototyping. We opted for jCOLIBRI2 (http://gaia.fdi.ucm.es/) to be used as the CBR framework in this study. jCOLIBRI2 uses Java development environment tool of Eclipse (http://eclipse.org/) as the programming language. The case base stored 24 cases. These real cases were acquired from the National University medical centre. They were represented using the attribute-value pairs technique. It consisted of a diabetic patient‘s dietary menu planning that was consulted by a dietitian. The nearest neighbourhood technique was used to retrieve the best case.

The adaptation of best case involves the ontology reasoning. The implementation of this task was made using Jena (http://jena.apache.org/), a framework of the semantic web application. Jena provides ARQ as their SPARQL query language for Resource Description Framework (RDF) data. Jena supports several reasoners i.e. RDFS, OWL, transitive and generic rule-based reasoner. In this study, we opted for the use of the OWL reasoner to support the instantiation and automated classification reasoning.

The process model of the proposed approach is shown in Fig. 1. A dietitian needs to input the patient’s details and their MET information. The patient’s details are divided into two parts i.e. patient’s background and food history. MET is a tool designed by a dietitian to guide the dietary menu planning according to the patient’s energy needs, which are measured in calories (kcal). It consists of eight food groups and five mealtimes. The total exchange portion of each food group is distributed according to the mealtime. Table 1 shows a sample of MET with 1800 calories. MET is the physical constraint that must be satisfied to ensure that a patient has a balanced diet each day. For example, during breakfast, the food items in the dietary menu planning must consist of starch, fruit, legume and fat food groups.

The final step in this study is the evaluation of the performance of the proposed approach by comparing them with the conventional approach. We conducted a survey among respondent to acquire their feedback.

The adaptation process begins once the best case is retrieved. It involves the satisfaction of multiple design constraints i.e. physical, preference and common sense. The explanation of how each constraint is satisfied through ontology reasoning is discussed in the subsection below.

A. Query and Inference Mechanism

Two elements in the ontology that were utilized to support the design case adaptation are querying and inferencing. Querying retrieves the asserted triples that match the given query. Inferencing, on the other hand, is the process of inferring new additional RDF statements from the asserted one. This process is computed automatically by a reasoner. In this research, two DL reasoning mechanisms have been exploited i.e. instantiation and automatic classification.

An instantiation checks whether an individual a is an instance of the concept C. This mechanism allows us to trace all the superclasses belonging to an instance. We used this mechanism to identify the food groups of the food items. For example, the immediate food group or class of brown rice is rice and rice products. However, in menu planning, we looked for the more general food group of brown rice i.e. starch. The instance checking inferred all the superclasses belonging to brown rice and found starches as one of its superclasses.

Automatic classification infers all members of the superclass from their subclasses. This inference mechanism was used when we wanted to compute the class hierarchy of the food group listed in Table 1.

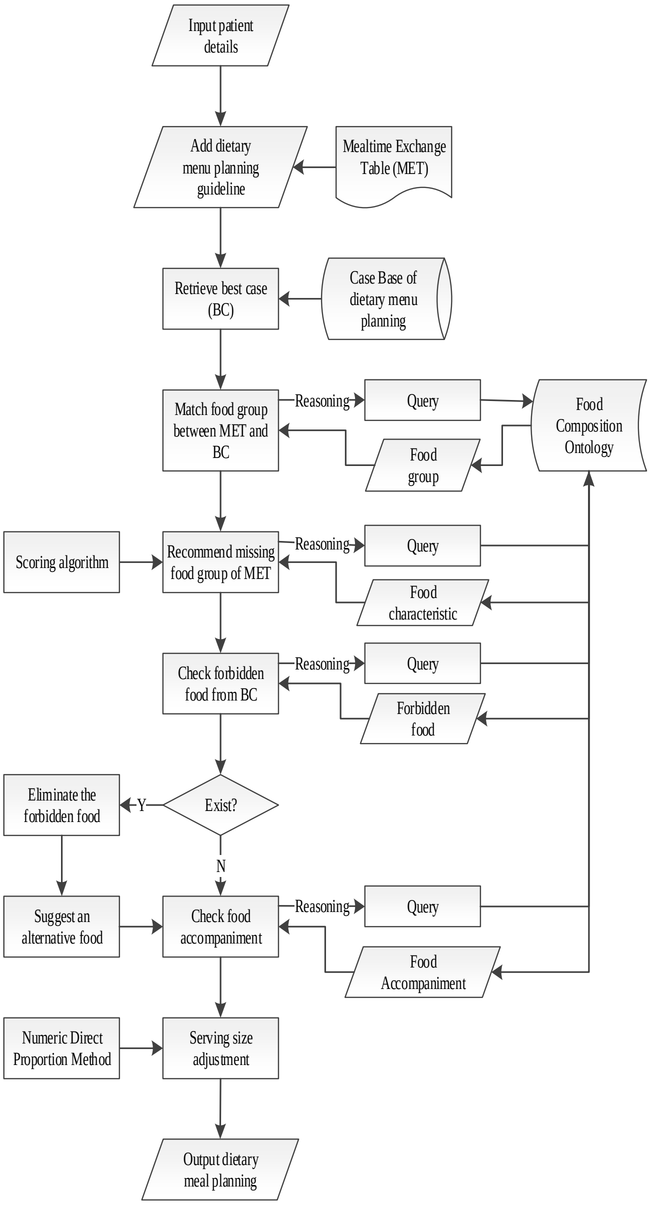
Input patient details
Add dietary menu planning guideline
Mealtime Exchange Table (MET)
Retrieve best case (BC)
Case Base of dietary menu planning
Match food group between MET and BC
Reasoning
Query
Food group
Food Composition Ontology
Scoring algorithm
Recommend missing food group of MET
Reasoning
Query
Food characteristic
Check forbidden food from BC
Reasoning
Query
Forbidden food
Eliminate the forbidden food
Y
Exist?
N
Suggest an alternative food
Check food accompaniment
Reasoning
Query
Food Accompaniment
Numeric Direct Proportion Method
Serving size adjustment
Output dietary meal planning

Fig. 1 The process model of the proposed approach

B. *The Matching Process*

The first physical constraint that must be satisfied is the food groups which have been allotted in MET. The design case adaptation involved is the matching process between the food groups in MET and the food items' food group in the best case at the same mealtime. If they are matched, then the food item will be recommended for the dietary menu planning. Here, instantiation plays the role where the food item in the best case has been inferred to identify which food group it belongs to. Fig. 2 shows the algorithm used in matching the food groups in MET with the food items' food group in the best case.

TABLE I
DISTRIBUTION OF EXCHANGES OF PORTION ACCORDING TO FOOD GROUPS AND MEALTIME FOR 1800 KCAL/ DAY

Food group	**Exc. portion**	**BF**	**MS**	**L**	**AS**	**D**
Starch	9	2	1	3		3
Vegetables	4			2		2
Fruits	3	1			1	1
Milk	2		1		1	
Seafood	4			4		
Meat	3					3
Legumes	1	1				
Fat	8	1		3	1	3

[Exc. = exchange, BF = breakfast, MS = morning snack, L = Lunch, AS = afternoon snack, D = dinner]

```
Loop by MET
  check if food group is in solution array (based
  on same mealtime and food group)
  if found
    check if item is cooked or raw food
    if cooked food
      check for all items combination food group
      exists in MET
  if not found
    recommend food item from ontology (call
    insert module)
  add into adapt array
```

Fig. 2 Algorithm for the matching process of food groups in MET

1) *The Missing Food Group*

If there is no match of the food groups between MET and the best case, it is then considered as the missing food group. At this time, the system will recommend a new food item for the missing food group. This operation involves the insertion of new elements for the design solution by using structural transformation adaptation. The ontology reasoning performs the automatic classification to generate the inferred class hierarchy of the missing food group. The individuals that are inferred to become members of the missing food group class are the candidates to be considered as a new food item recommendation.

The first sub-process of this task is to filter the food item candidates from the ineligible ones. For cooked foods, three criteria are needed to filter ineligible food items i.e. forbidden food; unavailable food group and exceed serving size for ingredients. For raw foods, filtration is performed on any forbidden food only. The algorithm of this sub-process is shown in Fig. 3.

The second sub-process involves healthy food choices for a diabetic with consideration of the patient's preferences. It starts with a second filter to separate a food item from the one that cannot be eaten based on two conditions. The first condition is whether food item in the food group can be eaten raw or must be cooked.

```
if cooked food
  check if each item combination is allergic or
  prohibited
    remove from item list
  check if there is one or more item
  combination does not exist in MET
    remove from item list
  check if there is one or more item
  combination exceeds the serving size
    remove from item list
if raw food
  check if food item is allergic or prohibited
    remove from item list
```

Fig. 3 Algorithm for ineligible food item filtration

The second condition is whether the food item can be eaten alone or not. From these conditions, the food groups are classified into three categories. The first category belongs to the food group that must be cooked and can be eaten alone, i.e. cereal-based dishes, vegetable dishes, legume-based dishes, seafood dishes and meat dishes. The second category belongs to the food group that can be eaten raw or must be cooked, and can be eaten alone or not. For any food item that cannot be eaten raw or eaten alone, it is assigned with the object property `isEdible` to false. Starches, vegetables, legume, and fats food groups fall under this category. For example, food items that cannot be the final candidate are brown rice from the starches food group since it cannot be eaten alone, kidney beans from the legume food group since it cannot be eaten raw, and peanut butter and low fat margarine from the fast food group because they cannot be eaten alone. The last category belongs to the food group that can be eaten raw and eaten alone, which are milk and fruits.

Next, the food selection criteria (FSC) are retrieved using the SPARQL query. These include the diabetes superfood, mealtime suitability, race cuisine, and nutrients that are significant to the food group that belongs to the food item. Among the nutrients are fibre for starches, vegetables, fruits and the legume food group; saturated fat and cholesterol for seafood, meats and fats food group; and monounsaturated (MUFA) and polyunsaturated fat (PUFA) for the fast food group. Using the OWL properties, the first three selection criteria are assigned under object property, while the nutrient values are assigned under data property.

Each of the food selection criteria has been assigned with its own weightage in order to calculate the score obtained by a food item. The total score of food item is obtained through the summation of similarity and the normalization of food selection criteria multiplied by its weightage. The first three selection criteria use the similarity concept, while the nutrient values use the normalization concept. The scoring algorithm is formulated using the following equation.

$$S = S_{sim} + S_{norm} \tag{1}$$

$$S_{sim} = \sum_{i=1}^{3} w_i \times sim_i \tag{2}$$

$$S_{norm} = \sum_{i=1}^{n} w_i \times norm_i \tag{3}$$

$$norm_i = x_i / \max(x) \tag{4}$$

Where,

$$sim_i = \begin{cases} 1, & input = food\ selection\ criteria \\ 0, & input \neq food\ selection\ criteria \end{cases}$$

$w_{i} = weigthage\ to\ the\ i^{th}\ food\ selection\ criteria$

$x_i = i^{th}\ nutrient\ value$

$\max(x) = maximum\ value\ of\ nutrient$

For example, the missing food group during breakfast is a legume. Dhal (yellow lentils) gravy is one of the candidates from this food group. To calculate the score for the dhal gravy, we start with the similarity function followed by the normalization. For the similarity function, input from the system is compared to the characteristics of the dhal gravy in food composition ontology. These characteristics are used as the criteria for food selection. The first criterion checks whether dhal gravy is a superfood for diabetes. Since dhal gravy is the superfood of diabetes, one point is assigned to this criterion. The second criterion checks whether dhal gravy is suitable to be taken during the missing mealtime i.e. breakfast. Dhal gravy has mealtime suitability ranging from breakfast until dinner. Due to this similarity, it gets another one point to this criterion. The third criterion is dependent on a patient's race. In this scenario, the patient's race is Indian. The system checks which race normally take dhal as their cuisine food. Dhal is normally taken by Indians according to [18], [19]. Due to this similarity, dhal gravy hits another one point.

The last criterion to consider is the nutrients that are significant to the food group. Fiber is a nutrient significant to the legume food group. Hence, normalization is applied to this criterion. First, the maximum value of this nutrient among all candidates in the legume food group is sought for. The maximum value is 7.8 which coincidentally belong to dhal gravy. The fiber value of dhal gravy is divided with the maximum nutrient value which equals to one. To calculate the total score gained by dhal gravy, each of the food selection criteria is multiplied with its own weightage. The weightage assigned for the food selection criteria, i.e. the diabetes superfood, mealtime suitability, race cuisine and nutrient values are 0.9, 1.7, 0.3 and 0.9 respectively. The following Fig. 4 illustrates the calculation of the dhal gravy score.

$$S_{dhalGravy} = 0.9\,(1) + 1.7(1) + 0.3(1) + 0.9(1) \;= 3.8$$

Fig. 4 The calculation of the dhal gravy score

The score for the other candidates from the legume food group is cooked chick-peas (3.32), baked beans (3.00), soy bean curd (1.50), and baked stuffed tofu (0.92). The food item with the highest score will be selected as the recommendation for the missing food group. Dhal gravy obtained the highest score compared to the other candidates. Thus it is selected to be recommended for the missing food group. By using this approach, the preference constraints have been satisfied. Fig. 5 shows the algorithm for the second subprocess in selecting the highest score obtained by a food item with consideration of the preferences factor.

By accomplishing this process, a complete dietary menu planning is generated. Next, another physical constraint is checked for forbidden food that exists in the best case, if any. This adaptation is required to ensure the dietary menu planning complies with the requirements of a new patient.

```
Check if foodItem foodgroup is CerealBasedDishes,
VegetablesDishes, LegumesDishes, MeatDishes,
SeafoodDishes

or foodgroup is Starches and hasEdibleRaw is True
or foodgroup is Vegetable and hasEdibleRaw is
True

or foodgroup is Legume and hasEdibleRaw is True or
foodgroup is Fats and hasEdibleRaw is True or
foodgroup is Milk, Fruits

   Calculate max value for Fibre, SaturatedFat,
   Cholesterol, MUFA, PUFA

   Calculate similarity

   Calculate normalization

      if foodgroup is Starches, Vegetables, Fruits
      or Legumes then calculate fibre

      if foodgroup is Meat, Seafood or Fats then
      calculate SaturatedFat and Cholesterol

      if foodgroup is Fats then calculate MUFA and
      PUFA
         Choose the highest score
```

Fig. 5 Algorithm for scoring calculation

C. *The Forbidden Food*

In this study, a forbidden food consists of allergy and/or prohibited food. The first type is due to health, while the latter is due to religious practice. The modifications of the normal diet involve the elimination of forbidden food and substituting it with item-for-item at meals [20]. This mechanism applies the deletion and replacement components under structural transformation adaptation.

A forbidden food is identified using the SPARQL query, where it retrieves the object properties `hasAllergic` and `hasProhibit` from each individual food item which contains the forbidden food. For example, peanut butter is assigned with `hasAllergic` peanut. Peanut is one of the food allergens. If the food item contains a forbidden food, it deletes and replaces the forbidden food using the category which it belongs to. Three categories of replacement are dependent if the forbidden food has accompaniment, item-for-item meal if the forbidden food can substitute with edible food in order to maintain the same meal's form and independent if the food item can be eaten alone.

For a dependent food, this study focuses on secondary food item which is an allergic food. The first step taken is to find an alternative for the secondary item that can be eaten together with the primary food. For example, a patient is allergic to peanuts. The dietary menu planning recommends wholemeal bread and peanut butter for breakfast. Since the patient is allergic to peanut, which is an allergen, peanut butter becomes the allergic food to be avoided. The system then tries to find an alternative accompaniment food for whole meal bread with the same food group with peanut butter. If there is an alternative, it will substitute the peanut butter. In this case, peanut butter is substituted with low-fat margarine. However, if there is no alternative for the secondary food item, both food items then need to be substituted since the primary food cannot be eaten alone.

Generally, the second category is likely to happen with cooked food. For example, the stir-fried bitter gourd is a vegetable dish. It contains dried anchovies as one of the main ingredients. For patients who have fish as their allergy, this means that they are not able to eat dried anchovies. The allergic food can be substituted with dried shrimps. Research done by the Food Allergy Research and Education (FARE) states that being allergic to finned fish does not mean that we cannot eat shellfish [21]. For any cooked food which an ingredient cannot be substituted, the system will recommend other cooked foods using the insert module that has been discussed in the subsection of the matching process above.

If the allergic food is independent, the substitution process is performed directly using the insert module. Fig. 6 below shows the algorithm for forbidden food.

```
Loop by forbidden list

   check food item is cooked or raw food

      if cooked food

         check each ingredient is not allergic
         and prohibited
```

```
    if raw food
      check food item is not allergic and
      prohibited
  if allergic or prohibited
    check if food item has hasServedWith
    property (dependent)
      change with alternative from the same
      food group and not allergic
    if no alternative
      change the primary food and food item
      with data from ontology (call insert
      module)
    check if food item has hasExchange property
    (item- for-item)
      change with alternative ingredient which
      is not allergic
    if no alternative
      change food item with data from ontology
      (call insert module)
if not dependent or item-for-item change food
item with data from ontology (call insert
module)
```

Fig. 6. Algorithm for forbidden food

D. *Food Accompaniment*

Dietary menu planning involves the common sense constraints from the context of food accompaniment. In this study, food accompaniment is divided into three categories. The first category is food that must be eaten together, where the relationship between foods is referred to as `hasServedWith`. For example, chapatti `hasServedWith` dhal gravy. The second category is the inverse of the first category, where the relationship between foods is referred to as `isServedWith`. However, not all foods from the first category can have the inverse relationship. For example, oatmeal `hasServedWith` milk, but milk is not compulsory to be eaten with oatmeal. The third category is food that has companion where it is advisable to be eaten together, and the relationship between them is referred to as `hasCompanion`. For instance, oatmeal `hasCompanion` with banana.

The accompaniments of a food item are identified using the SPARQL query. A food item which has the accompaniment relationship creates the accompaniment array. This array does not have any food accompaniment that is forbidden to a patient. Here, we refer to the food item as the primary food item, while the food accompaniments are referred to as the secondary food item. The system then checks whether the food group of food accompaniments exist in MET at the same mealtime. If the food group does not exist, the food item needs to be substituted since it cannot be eaten without any accompaniment. This is only applicable for `hasServedWith` and `isServedWith` relationships.

To ensure that the food item has an appropriate food accompaniment, the system checks for the existence of food accompaniment. First, it checks whether the food accompaniment exists in the adapt list. If it already exists, then no further action is involved. However, if the accompaniment food does not exist in the adapt list, the system checks whether it exists in the best case list. If it already exists, the system adds the food accompaniment into the adapt list. If the food accompaniment does not exist in the best case, the system recommends a new food accompaniment. The candidates come from the accompaniment list. The system uses the insert module in selecting the best food accompaniment. Fig. 7 shows the food accompaniment algorithm.

```
Loop by adapt List
  Check if food item has hasServedWith or
  isServedWith or hasCompanion properties
    Retrieve all available food accompaniments
      (accompaniment array) which are not
    forbidden
      Check if each food group of food
      accompaniment exists in MET on same
      mealtime
      If not exist and hasServedWith and
      isServedwith  property
        substitute food item (call insert
        module)
      Check if food accompaniment already exists
      in adapt list
        no change
      Check if food accompaniment already exists
      in best  case
        select food accompaniment
If not exist, recommend new food accompaniment
which exists in accompaniment list (call insert
module to select the best one)
```

Fig. 7 Algorithm for food accompaniment

III. Result and Discussion

A comparison between the conventional and the proposed approach was executed to evaluate the performance of the proposed approach. The conventional approach used case combination technique to design case adaptation. The evaluation measured the suitability of the dietary menu planning generated by ontology and case combination in satisfying the nutritional constraints and aesthetic criteria i.e. food preferences and accompaniment. It was evaluated by 10 postgraduate students of the dietetic program. Each student was asked to complete a scenario-based questionnaire. Three sets of scenarios were prepared to cover the different nutritional needs and preference constraints of the patients.

Evaluation criteria were divided into three sections. The first section intends as to evaluate the ontological and case combination approach to fulfilling the physical constraints. This can be measure by checking whether the food items in dietary menu planning match with the food group in MET. Another physical constraint is the forbidden food. It consists of prohibited and allergy food. A respondent has to check that the dietary menu planning does not contain any of prohibit and/ or allergy food(s) belong to a patient. The second section evaluates the personal preferences constraint. This constraint is reflecting in patient's cultural food customs. This criterion is measured by checking whether the recommended menu consists of food item that is normally eaten by the patient's race. The third section evaluates the inconsistency of solutions. This issue is measured by the relationship of food accompaniment in dietary menu planning context. The first and second sections of the evaluation criteria are regard with the second objective. And, the third section is regard with the third objective. The first objective is to evaluate indirectly by its ability to fulfil the physical and preference constraints; and consistency in food accompaniment.

Overall feedback solicited shows that the proposed approach has better and equal performance compared to the conventional approach in designing the dietary menu planning. In supporting design tasks which require physical, preference and aesthetic constraints to be met, ontological approach capable of accomplishing all of these constraints. On the other hand, case combination has the same capability to meet the physical constraints but not very good in coping with personal preferences and aesthetic constraints.

IV. Conclusions

In this paper, an ontological approach was developed to perform design case adaptation. We applied the proposed approach to a dietary menu planning domain. The ontology supported both the transformation and substitution adaptation tasks. It was able to satisfy the physical, preference and common sense constraints using the adaptation tasks above. DL inference mechanisms i.e. instantiation and automatic classification were used to infer the new (additional) triples. SPARQL retrieved the inferred and asserted triples that matched a given query.

For future works, we plan to apply the ontological case adaptation algorithm in other CBR synthetic tasks such as planning and configuration.

References

[1] J. Kolodner, Case-Based Reasoning. Morgan Kaufmann, 1993.

[2] A. Gómez de Silva Garza, "An Evolutionary Approach to Design Case Adaptation," University of Sydney, 2000.

[3] J. L. Kolodner and D. B. Leake, "A Tutorial Introduction to Case-Based Reasoning," in Case-Based Reasoning: Experiences, Lessons and Future Direction, D. B. Leake, Ed. Menlo Park, CA: American Association for Artificial Intelligence, 1996, pp. 31–66.

[4] C. R. Marling, G. J. Petot, and L. S. Sterling, "Integrating Case-based and Rule-based Reasoning to Meet Multiple Design Constraints," Computational Intelligence, vol. 15, no. 3, pp. 308–332, 1999.

[5] F. Gebhart, A. Vob, W. Grather, and B. Schmidt-Belz, Reasoning with Complex Cases. AH Dordrecht: Kluwer Academic Publishers Group, 1997.

[6] P. Pu and L. Purvis, "Formalizing the Adaptation Process for Case-based Design," in Issues and Applications of Case-Based Reasoning in Design, M. Lou Maher and Pear, Eds. New Jersey: Lawrence Erlbaum Associates, 1997, pp. 221–240.

[7] M. Lou Maher, "CASECAD and CADSYN," in Issues and Applications of Case-Based Reasoning in Design, M. Lou Maher and P. Pu, Eds. Mahwah, New Jersey: Lawrence Erlbaum Associates, Inc, 1997, p. 335.

[8] B. Faltings, "Case Reuse by Model-based Interpretation," in Issues and Applications of Case-Based Reasoning in Design, 1st ed., M. Lou Maher and P. Pu, Eds. New Jersey: Lawrence Erlbaum Associates, 1997, pp. 39–60.

[9] B. Medjdoub, "Constraint-Based Adaptation For Complex Space Configuration In Building Services," Journal of Information Technology in Construction, vol. 14, pp. 724–735, 2009.

[10] S. Ono, T. Izumi, A. Fujiyama, C. J. Ashley, and S. Nakayama, "Interior Coordination Using Case-Based Reasoning and Constraint Satisfaction Paradigm," in lEEE Conference on Cybernetics and Intelligent Systems, 2004, pp. 1067–1072.

[11] A. Gómez de Silva Garza and M. Lou Maher, "GENCAD: A Hybrid Analogical/Evolutionary Model of Creative Design," in 5th Intl. Conf. on Computational and Cognitive Models of Creative Design, 2001, pp. 1–32.

[12] M. d'Aquin, J. Lieber, and A. Napoli, "Decentralized case-based reasoning and Semantic Web technologies applied to decision support in oncology," Knowledge Engineering Review, vol. 28, no. 4, pp. 425–449, 2013.

[13] J. A. Recio-garcia, B. Diaz-Agudo, P. Gonzalez-Calero, and A. Sanchez-Ruiz-Granados, "Ontology based CBR with jCOLIBRI," in Proceedings of AI-2006,the Twenty-sixth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, 2006, pp. 149–162.

[14] S. Shaharin, A. Saad, M. Hashim, and H. Ubaidullah, "Current Trends In The Integration Of Case-Based Reasoning And Semantic Web," no. May. p. 13, 2016.

[15] N. F. Noy and D. L. McGuinness, "Ontology Development 101 : A Guide to Creating Your First Ontology," 2001.

[16] N. Yusof, S. A. Noah, and S. T. Wahid, "Ontology Modelling of Malaysian Food Exchange List," in 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, 2015, vol. 2: KEOD, no. Ic3k, pp. 301–306.

[17] N. Yusof, S. A. Noah, and S. T. Wahid, “Ontology Modeling of Malaysian Food Composition,” in Third International Conference on Information Retrieval and Knowledge Management (CAMP’ 16), 2016, pp. 149–154.

[18] “Malaysian Indian cuisine Mamak culture,” Wikipedia, 2015. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Malaysian_Indian_cuisine&oldid=661144060. [Accessed: 26-May-2015].

[19] -, “What is Malaysian Indian Food,” MalaysianFood.net, 2013. [Online]. Available: http://www.malaysianfood.net/Indianfood.html. [Accessed: 08-Nov-2015].

[20] K. L. Mahan, S. Escott-Stump, and J. L. Raymond, Krause’s Food & The Nutrition Care Process, 13th ed. Elsevier, 2012.

[21] “Fish Allergy,” 2014. [Online]. Available: http://www.foodallergy.org/. [Accessed: 25-Apr-2015].