Exploiting knowledge about fashion to provide personalised clothing recommendations

Dimitrios Vogiatzis a,∗, Dimitrios Pierrakos a, Georgios Paliouras a and Sue Jenkyn-Jones b

a Institute of Informatics & Telecommunications, NCSR “Demokritos”,
Ag. Paraskevi 15310, Athens, Greece
E-mail: {dimitrv,dpie,paliourg}@iit.demokritos.gr
b London College of Fashion, London, UK
E-mail: s.jenkyn-jones@fashion.arts.ac.uk

Abstract. In the fashion industry, mass-customisation is a new trend that tries to produce clothes respecting the idiosyncracy of every customer and doing so cost effectively. In this paper we present a knowledge framework that leverages the above process by providing personalised clothing recommendations. The methodology that we propose, and the prototype we have built, incorporates knowledge about aspects of fashion, such as materials, garments, colours, body types etc. into an ontology. With the aid of concepts and relations of the ontology, domain experts can also define style advice rules. Moreover, a general-purpose personalisation server (PServer) is employed, that stores style advice rules in the form of user stereotypes and mines user interaction data to produce patterns that enrich the experts' style advice rules. Due to the synergy of the domain ontology and the PServer there is an impetus to map style advice rules between the two different representations. Finally, a recommendation engine that exploits user stereotypes is built in order to suggest new fashion items to users.

Keywords: Fashion Ontology, User Modeling, Personalisation

1. Introduction

The fashion domain is a field of business where the requirement for adaptation of products to interests, needs, and personal physical characteristics (such as body type) of customers is very high. Personalisation in the fashion domain is the tool to achieve adaption, and as such it is considered important and adds value to the services provided.

Fashion experts have established some guidelines (albeit fluid occasionally) about appropriate style, fit of garments for different occasions, different body types, facial features etc. Moreover, fashion oriented web sites or social networking sites, collect user transactions regarding expression of preferences, or purchase of garments. Thus, there are two types information resources available: explicit style advice rules, and data denoting preferences. Both resources are useful for retailers aiming to achieve a competitive advantage. The first type, i.e. that of style advice rules, can be eventually represented in a formal form, such as that provided by an ontology. The latter, is usually mined so that important patterns are discovered that denote general user tendencies. Eventually, both types of information will co-exist and will be used to provide personalised information to the user.

An important issue in dealing with expert knowledge, is the organization and handling of the unstructured information that characterizes the fashion domain. In particular, information in this domain comes from various sources, such as the manufacturers, the human morphology, the garment styles, the occasion

∗Corresponding author. E-mail: dimitrv@iit.demokritos.gr.
for a garment, etc. Although, there are associations between the above types of information sources (in the form of loose guidelines), they are not usually expressed in a way that could form a fashion oriented knowledge base. Thus, there is the requirement for processing this information, extract the available knowledge and present it in a more structured and manageable form.

It is also necessary, to provide the knowledge infrastructure for representing user models pertinent to the fashion domain. The user models, among other things, should contain personal information such as body measurements, body types, age, facial features but also garment preferences or garment appropriateness.

The end product of personalisation is garment recommendation, which aims to assist the user in getting quickly to the information that the customer is seeking, as well as to provide the user with alternative product options.

**Contribution of our work**

In this paper, we present an ontology-based recommendation system, for style advice in the fashion domain. The proposed system, handles both domain expertise, and user transaction data, and consists of two main components: a semantic knowledge repository, and a general-purpose personalisation server, named PServer.

The semantic knowledge repository is a fashion ontology based on OWL, which incorporates structures for representing humans as required by the fashion domain (i.e., body measurements, body types, facial features, etc.), clothes, and materials—all this is provided by domain experts. Moreover, the knowledge structure, contains two levels of rules. The first level named attribute rules, maps measurements to intermediate concepts (e.g. body measurements to body types). The intermediate concepts are named attributes. The second level of rules, named style advice rules, maps attributes to garment or colour advice (e.g. the average body type is recommended a pleated skirt type).

PServer acts as repository for individual user models, which are created as the users interact with the system. It also operates as a repository for the aforementioned style advice rules, which are represented as user stereotypes. PServer’s main strength is the statistic processing of user transactional data to update user models, and subsequently figure out whether the the style advice rules conform to users’ preferences. Alternatively, our work within the SERVIVE project, which aims to create an infrastructure for mass-customisation in the fashion industry \(^1\), is to combine the strength of expert advice with statistical data analysis. This is achieved by the ontology-based recommendation system.

The rest of this paper is organized as follows. In section 2 we review ontologies and style advice systems realted to the fashion domain. Then in section 3 we present the ontology developed in the context of this project. In section 4 we present PServer and in section 5 we define a mapping from OWL to PServer stereotypes, which are used in the personalisation. Next, in section 6 we explain in detail the architecture of the proposed system that encompasses the OWL and PServer. Conclusions are drawn in section 7.

## 2. Background Knowledge & Related Work

### Ontologies in the Fashion domain

A number of attempts have been made that tried to build ontological representations for the fashion domain. BodyXML was such an attempt, that specified a European “XML wrapper” standard that integrated body and product representations \([4]\). In BodyXML, a “person” is a unique individual. The person has two attributes: “details” - unique information such as their name and contract information, colour of hair, eyes and skin, shopping and colour preferences etc; and multiple “representation(s)” - each of which could be a picture of a body part, a point cloud from a scan, a set of body measurements, etc. In BodyXML, a “product” refers to a clothing item. The product again has two attributes: “details” — unique information such as retailer and manufacturer names, textiles, care instructions etc; and multiple “representation(s)” - each of which might be a picture of a specific garment, such as a size 8 red garment, and its measurements etc.

Another attempt produced ontology for garments \([2]\). This ontology included an abstract description of the human body and an ontology for clothing patterns. The ontology introduced a “garments” class for single pieces of garments, which can be defined as a collection of clothing patterns that are sewn together. Commonly used descriptions of garments such as jackets, trousers, skirts and dresses can be introduced as subclasses of garment class.

---

\(^1\) www.servive.eu
However, the above approaches are limited in terms of knowledge representation since they cover only a small part of the available knowledge in the fashion and human body domains. A more detailed representation of the concepts and their relationships is required. The SERVIVE fashion ontology (SFO) provides such a structured and unified vocabulary to represent humans, and garments.

**Stereotypes and user modeling**

User modeling technology aims to make information systems user-friendly, by adapting the behavior of the system to the needs of the individual. A user model primarily contains information that characterize the interaction of the user with the system and other users, provided interaction between users is supported [9]. In the fashion domain each personal user model consists of personal information such as age, body type, etc and style preferences. The personal information is represented by the *attributes* of the user model, whilst the style preferences correspond to user model *features*. An example of user model is depicted in table 1, where the numbers denote degree of user preference.

One of the earliest types of user model is the stereotype [10]. Stereotypes are collective user models and similarly to personal user models consist of two types of information. The *stereotype attributes* represent knowledge external to the application, usually personal things, such as body type, age, level of expertise in a domain etc. The *stereotype features*, refer to entities from within the application, such as garment types. A stereotype can be interpreted as “users with certain attribute values, are recommended or prefer certain features”. For example, the stereotype in table 2 says that users of average body type are highly recommended pleated skirts, but the recommendation for military jackets is medium; alternatively the same table can be interpreted as denoting preferences.

Yet another type of collective user model is the community, which can be thought of as a set of users having similar preferences for features of the application. Table 3 presents a community of users preferring pleated skirts and jackets. Communities can be used in personalisation, for example if it could be inferred that a user who likes pleated skirts also likes jackets. Communities, are usually produced with data mining algorithms when applied to users’ transaction data.

Stereotypes are central to our work, in implementing style advice. The attributes of an (atomic) model, will be compared against the attributes of available stereotype types, seeking a possible match. The features and feature values of the matching stereotypes form the style advice. It is quite possible, especially for a new user, not to have yet any features values. However if many users with the same attributes values have interacted with the system, it is interesting to figure out whether their average feature preferences, conform to those of the expert provided stereotype. For example, the experts say that average body type should wear pleated skirts, but not peplum jacket; but what are the customers’ view on that?

**Recommendation Systems**

A popular application of user modeling technology is in the **Recommender systems (RS)**, which have evolved especially in the Web. They describe any system that produces recommendations or guides the user in a personalised way to interesting or useful objects in a large space of possible options [1]. RS are usually classified into the following categories, based on how recommendations are formed:

- **Content-based recommendations**: The user will be recommended items similar to the ones he preferred in the past.
- **Collaborative recommendations**: The user will be recommended items that people with similar tastes and preferences liked in the past.
- **Hybrid approaches**: Combination of collaborative and content-based methods.

Recommender systems have been used in a variety of e-commerce applications in the fashion domain.
These applications exploit a number of parameters to provide recommendations, such as personal information (age, gender, height, weight), as well as information related to style, fashion trends, etc. Example applications are:

**Levis Style Finder**, which provides recommendations related to the company’s garments. Each customer submits to the application gender information and subsequently rates a set of product categories. The system follows a collaborative filtering approach to provide recommendations.

**What am I Gonna Wear**, where each user creates his/her wardrobe that includes the garment categories according to his/her preferences. A recommender system is used to suggest to the user garments similar to the wardrobe’s garments, i.e., following a content-based approach, or garments that have been chosen by similar users, i.e., through a collaborative filtering approach.

**MyShoppingPal.com**, where each user submits information about his/her body type and his/her style preferences. This information is used for garment suggestion. A similar system is the **MyShape** recommendation system.

**My Virtual Model**, where each customer “dresses” his/her virtual model, based on the body type and the style preferences. Subsequently the recommendation system suggests to the customer garments that fit this virtual model.

In yet another approach, ontologies are used in conjunction with recommender systems. A popular method is to enhance the user’s interest profile, with ontological concepts that are close to the user’s expressed interests. For instance, the user’s interest to a concept can be propagated to the super-concept, which can be integrated into the user’s interest profile [7].

**Knowledge Based Systems**

Knowledge based systems represent fashion style advice in the form of *if-then* rules, either crisp or fuzzy, which encode expert domain knowledge. **Shirt-MC** is an expert system for the mass customisation of garments, and in particular of shirts [6]. In that system, the users provide information about height, weight, complexion as well as some subjective pieces of information. Then the user receives feedback, and there is a second range of user options concerning the trendiness and the freshness of the garment. The knowledge of the system is encoded in an expert data base. In another approach, the aim is to suggest matching or *coordinated* clothing items. This is tackled at two levels. First, the relevant attributes of clothes are detected and recorded by experts. Then a rule based system is built to evaluate the coordination degree of pairs of clothing attributes. At the second level, the coordination levels of clothing attributes are combined to provide the coordination degree (how well they fit) of whole garments. At this level a TG fuzzy neural network is employed [13].

All the above fashion recommendation systems are based on simple user models which are created ad-hoc and they exploit information provided by customers in an unstructured manner. In contrast to the above approaches, the system presented in this work, provides a generic framework for providing recommendations in the fashion domain.

### 3. Knowledge Integration: Servive Fashion Ontology

Knowledge in the fashion domain comes from various sources. Knowledge about garments and the various human styles, usually provided by the manufacturers and fashion experts of the garments, as well as human morphology. Although there are associations between the above types of information source, the majority of knowledge that this information conveys is difficult to be managed efficiently. Thus, there is the requirement for processing this information, extracting the available knowledge and presenting it in a more structured and manageable form.

The **Servive Fashion Ontology (SFO)** provides a structured and unified vocabulary to represent human, fashion and manufacturing concepts. The ontology shares a number of common terms and concepts from the above domains and it is further specialised to cover the needs of each part.

The SFO was developed in OWL 2.0 [3], with the aid of the Protégé 4.1 ontology editor. The ontology captures the experience of a style style advisor, which could be stated in abstract terms as: given some *body measurements* and some *facial features* infer the body type, and subsequently suggest a *garment type*, as well as some *garment colours*. In the current work, garments concern women’s clothes, and they fall into the following categories: knitwear, skirts, jackets, and two-piece business suits. SFO it is not meant to be a final and complete ontology, but an ongoing effort. As
such, it will be publicly available at the project web site\(^2\).

The main concepts in the ontology are humans and garments. Concerning humans, the related concepts are the body types, and facial features, such as skin colour, hair colour, eyes and eyebrows colour. Garments could be split into categories, such as garments for the top part, lower part or the whole of the human body. Also, there is the concept of garment material.

Given the above information, we have formed a number of classes, object and data properties to encode the experts’ knowledge. The top level of the ontology is depicted in figure 1. Henceforth, class names will have their first letter capitalised, individuals’ names will start will lower case, and property names start with the trigram "has".

- class:BodyType. The class represents the concept of Human Body Type, i.e., the general shape of a human. Based on the concept of “shape modeling driven by products”, the human shape is described by its LowerPart, which corresponds to the body part from the waist to the legs, the UpperPart, which corresponds to the body part from the waist to the head and the OverallBody which corresponds to the human body shape as a whole. Eight different body shapes implement the OverallBody concept, NormalFigure, BroadAtTop, BroadAtBottom, HourglassCurvy, OvalOverall, Narrow And Straight, BroadAndStraight and Atypical. These concepts are defined as subclasses of the OverallBody concept.

- class:Colours. The class models the various colours that exist in human, fashion and garment domains. These colours are modeled as class members (individuals). Examples of these individuals are the navy, black, blue, grey, etc.

- class:Garment. This is the main class that models the domain of clothes. It has three main sub-classes that refer to the garment suited for the top part, the bottom part and for the whole of the human body respectively.

- class:GarmentFeatures. The class models the characteristics that can be used to distinguish the various types of garment, such as Buttons, Cut, Colour, Pattern, etc. These characteristics correspond to the subclasses of the class.

- class:GarmentMaterial. The class models the various types of fabric that are used to produce the garment. The different types of fabric, i.e., Cotton, Linen, Silk, Wool, etc., correspond to the subclasses of the class.

- class:HumanColour. The class models the colours of a person, as described in the fashion domain. It has the following subclasses: EyesColour, EyebrowsColour, HairColour, and SkinColour. These parameters correspond to subclasses of the HumanColour class.

- class:HumanStyleColour. The class models the categories that a human can be assigned, based on the Season Analysis Model [5]. There are four different categories of human style that are represented as subclass of Human Style Colour class, i.e., Spring, Summer, Autumn and Winter.

- class:Occasion. The class Occasion models the cases that a human would select a particular garment. These cases, such as Workwear, or Sportswear, are the subclasses of the Occasion class.

- class:Style. The class models the various types of style that can be exploited to classify a human based on his/her dressing habits. These types can be Casual, Eclectic, etc., corresponding to the subclasses of the Style class.

- class:Human. The class models the human in the fashion domain. The individuals of this class are the “real” humans that are exploiting information in the fashion domain.

The relations between the instances of classes are modeled using Object Properties. Examples of Object properties are the following:

- hasEyesColour This is an object property with Human as domain, and range within the class EyesColour. Similarly, we have defined the object properties hasHairColour, hasEyeBrowsColour, hasSkinColour.

- hasHighRec. The property is an object type property of the Human class and has a range within the Garments. Its purpose is to associate humans with highly recommended garments. Similarly we have defined hasLowRec and hasNegRec object properties, to express low and negative recommendations respectively.

Finally, data properties correspond to the parameters from the body shape analysis, as well as human char-

\(^2\)http://www.servive.eu/
characteristics such as height, age, etc. Examples of data properties are described below:

- **hasHeight**, which is a property of the Human class, corresponding to human’s height.
- **hasBMI**, which is a property of the Human class, corresponding to the Body Mass Index of a human.
- **hasAge**, which is a property corresponding to the age of a human.

3.1. Rules and Reasoning

Apart from the representation of the main concepts and their relationships, SFO also represents rules for inferencing within the fashion domain. These rules fall into two types: (a) first level or attribute rules and (b) second level or style advice rules. The first type of rules associates human characteristics with higher-level concepts (named attributes), whilst the second type associates higher-level concepts with garment types or garment colours. The two types of rule are meant to work in conjunction; they are defined by fashion experts after examining various parameters, such as the available garment types, current fashion trends, age groups, etc. A description of these types of rules in Manchester syntax is given below:

**Attribute Rules (first-level rules)**

Attribute rules are created by fashion experts to denote relations between characteristics of humans that are modeled in the ontology. For example, they might associate human facial features with “human style colours”, i.e. summer, spring, autumn and winter.

```
Spring EquivalentTo
hasEyeBrowColour some {Light, DarkBlonde, GoldenNaturalBlonde} and
hasEyeColour some {Aqua, Hazelnut, Green, Golden, LightBrown} and
hasHairColour some {Light, DarkBlonde, GoldenNaturalBlonde} and
hasSkinColour some {Light, Frekles, Golden}
```

Another set of attribute rules allows the specification of a body type based on body measurements, which involves the usage of data properties.

```
OvalBodyType EquivalentTo
hasWaistHeight exactly 1 (float[<=100.825]) and
hasHeight exactly 1 (float[>"68"^^integer])
```

Note that the values of object and data properties are expressed in OWL as class descriptions with property restrictions. Moreover an inference engine supporting OWL-2.0, such as PELLET can be used to fire the first or second-level rules [11].

**Style advice rules (second level rules)**

The style advice rules are also defined by fashion experts and are build to relate intermediate concepts with garment characteristics, but also to denote the degree of association. In OWL, we represent the style advice rules as class definitions, for example, given the NormalFigure body type, the
hasHighRec object property, and the jacketFitted and militaryStyles individuals, the following specify what is highly appropriate and inappropriate for a human of normal body type:

NormalFigure EquivalentTo
   hasHighRec value jacketFitted

NormalFigure EquivalentTo
   hasNegRec value militaryStyles

Given the above definition, and marry being a member of the NormalFigure class it is inferred that,

(marry hasHighRec jacketFitted)

Another type of a style advice rule is the association of human colour style or “seasons” with garment colours. In the following example all users of the “Spring” type are recommended light, warm and bright colours for garments.

Spring EquivalentTo
   hasHighRec some {light, warm, bright}

Ontology Individuals

Using personal characteristics, an individual is created in the ontology with certain values on object properties. These values correspond to class instances.

More formally, let $A$, the set of user characteristics and $B$ the set of user characteristics values. Moreover, let $O$, the set of ontology elements, such as classes, object properties and data properties and $I$ the set of class instances. We define a “1-1” relation $\mathcal{R}$, that maps each pair $(a, b)$, where $a \in A$ and $b \in B$, onto the pair $(o, i)$, where $o \in O$ and $i \in I$, i.e.,

$$
\mathcal{R} : (A, B) \rightarrow (O, I) :
\mathcal{R}(a, b) = \{(o, i) | o \in O' \subseteq O, i \in I \text{ if } b \in \mathcal{R} \}.
$$

where $O'$ and $O''$ the sets of data properties and object properties of the ontology respectively. The created individual can be considered as a set of $(o, i)$ pairs.

As a concrete example, the user characteristic defined by the pair (skin colour, golden) is mapped onto the ontology’s object property (hasSkinColour,Golden), where “hasSkinColour” is an object property and “Golden” is an instance of the class “Colour”.

4. PServer

PServer is a general-purpose personalisation engine under development at NCSR "Demokritos". It has been used for personalisation in a variety of fields [8]. PServer operates as a Web Service, accepting http requests and returning XML documents with the results. Moreover, it can be used by many different applications concurrently. Any developer who needs to add personalisation to an application, is required to add a minimal amount of code for making the application a client of Pserver. Thus, PServer greatly facilitates the personalisation of existing applications.

PServer separates user modelling from the rest of the application and features a flexible, domain-independent data model that is based on four entities: users, that are represented by some identifier, attributes, that represent persistent user-dependent characteristics, features, that are application-dependent characteristics, which may or may not attract user preference and user models. PServer offers three types of user model: personal, stereotypes and communities. Moreover, PServer provides the option of exploit user interactions with the system and in particular, frequency counts and/or histories of actions in order to update the features of personal user models and user stereotypes. In this manner, we can at any point in time infer the level of interestingness of each user in a certain feature.

5. Using the fashion ontology for personalisation

For the purpose of personalisation, style advice (or second level) rules of the SFO are stored in the PServer. This is described next with concrete examples, and summarised in table 4.

SFO Ontology classes are mapped to PServer attributes. These classes refer to human body characteristics, such as body type, age or facial features.

$$
\text{class} : \text{BodyType} \rightarrow \text{attr} : \text{BodyType}
$$

Other classes are mapped to PServer features. They refer to garment features, such as skirt.pleated, jacket.military, etc.

$$
\text{subclass} : \text{Jacket.Military} \rightarrow \text{ftr} : \text{Jacket.Military}
$$

3http://www.iit.demokritos.gr/skel/
Table 4

<table>
<thead>
<tr>
<th>OWL</th>
<th>PServer</th>
</tr>
</thead>
<tbody>
<tr>
<td>class</td>
<td>Attribute</td>
</tr>
<tr>
<td></td>
<td>Feature</td>
</tr>
<tr>
<td></td>
<td>Attribute Value</td>
</tr>
<tr>
<td>equivalent classes</td>
<td>Stereotype</td>
</tr>
<tr>
<td>ObjProperty:hasHighRecom</td>
<td>1</td>
</tr>
<tr>
<td>ObjProperty:hasLowRecom</td>
<td>0.5</td>
</tr>
<tr>
<td>ObjProperty:hasNegRecom</td>
<td>-1</td>
</tr>
<tr>
<td>DataTypeProperty</td>
<td>Attribute</td>
</tr>
<tr>
<td>subClassOf:BodyType</td>
<td>Attribute Value</td>
</tr>
<tr>
<td>Individual:garment</td>
<td>Garment Feature</td>
</tr>
</tbody>
</table>

Finally, some classes are mapped to PServer attribute values, e.g. the class NormalFigure,

\[ \text{subclass : NormalFigure} \rightarrow \text{attr:Value : NormalFigure} \]

Data type properties can be mapped to a PServer attributes, e.g.,

\[ \text{dataTypePropert : hasAge} \rightarrow \text{attr : age} \]

A reasoner can be employed to derive attribute rules (these are the first-level rules). In particular it can assign a certain individual to an attribute rule. For instance the individual:

\[ u=\{\text{hasEyeBrowsColour,Light}, \text{hasEyeColour,Aqua}, \text{hasHairColour,Light}, \text{hasSkinColour,Golden}\} \]

can be mapped to the attribute rule \( \text{Spring} \).

Finally, given the following OWL construct expressing a style advice rule,

\[ \text{Oval EquivalentTo hasHighRec value jacketLoose} \]

it will be mapped to the following PServer stereotype,

\[ \text{attr : BodyType = Oval, ftr : jacket.loose = 1} \]

In other words, the OWL object property: highRec, midRec, lowRec will be mapped to PServer features values such as 1.0, 0.5, -1.0. The body type class Oval, becomes a PServer attribute value, and the OWL individual jacketLoose, is mapped to the PServer feature jacket.loose.

Apart from the aforementioned advice rules, that are stored in PServer as stereotypes (let us call them rigid), we can obtain another type of stereotype with the aid of PServer. These stereotypes, called flexible, reflect user preferences, as obtained through user interaction with the system. Flexible stereotypes, can be initialised from the rigid ones. Subsequently, atomic user models are processed to infer statistics about user preferences, and then to update the flexible stereotypes. Finally, flexible and rigid stereotypes can be combined to provide recommendations. For example, given the rigid stereotype: “Spring people are advised to purchase garments with light colours” (mentioned above), we might discover that that “Spring people” actually buy dark clothes. This, seeming contradiction might enrich the system: The user is given the choice of expert or peer advice.

6. System Architecture

SFO ontology and the PServer have been integrated into a system that is able to provide recommendation functionality for garments. The system, named Servive Style Advisor is a domain knowledge-based mechanism, continuously capturing consumer knowledge (preferences, individual customers design “creativity”), while guiding consumers in the making of their clothes. Style Advisor enables the accumulation and intelligent retrieval of knowledge acquired from prominent field experts, while at the same time the knowledge base is continuously adapted to customer preferences and buying styles, classified according to well defined customer groups (stereotypes).

Style Advisor consists of the following modules:

1. **The User Interface**, which is responsible for all the communication between the customer and the system.
2. **The Matching Stereotype engine**, which is connected to the PServer and extracts the stereotype of the particular customer.
3. **The Recommendation engine**, which generates the recommendations for each customer based on her stereotypes.
4. **The Knowledge Repository and Reasoner Engine** which corresponds to the storage of the domain’s semantic information, i.e., the ontology and offers the inference functionality via the PELLET reasoner.
5. The Manager Engine, responsible for the management of the system functionalities and the communication between the system modules.

The functionality of the Advisor consists of the following steps, depicted also in Figure 2:

- **User Mapping to Ontology.** The first step of the process is the association of a particular user to the classes and relations of the OWL ontology. A set of user characteristics, such as her “skin color” or “hair color” are submitted to the application through user interface and an individual for the Human class is created in the ontology. As a concrete example, in Figure 2, we can see that an individual named Mary submits via the user interface her skin and hair colors respectively, in our scenario light and blonde. Subsequently, the system creates an individual in the ontology, named Mary, and assigns these characteristics to the corresponding object properties of the ontology, i.e., “hasSkinColour=light” and “hasHairColor=blonde”.

- **Extracting Attribute Rules.** Using PELLET and the attribute rules, the particular individual is also assigned to an equivalent class. For instance, the individual Mary with the object properties described above, can be assigned to the class that describes her *Style Colour*, which in Mary’s case is “Spring”.

- **PServer Stereotype Retrieval.** Having inferred the individual’s attributes, the PServer’s stereotypes are triggered. This step is realized by initially assigning to the PServer’s attribute values the corresponding ontology class, as represented by the attribute rule and subsequently retrieving the stereotypes which correspond to the appropriate attribute values. The triggered stereotypes are represented by a set of features which are unique for the particular stereotypes. In the fashion domain that we exploit, the stereotype features describe the various characteristics of the garments, such as color, material, style, etc. Using Mary’s case, the class (attribute rule) “Spring” is mapped to the attribute “human color style”, with the attribute value “Spring”, which triggers the stereotype “SpringStereotype”. The features of the stereotype are “jacket.loose=1, jacket.color.azur=1, ...”. The above process is depicted below:

\[
\text{attr.rule} : \text{Spring} \rightarrow \text{attr} : \text{Spring} \\
\text{attr} : \text{Spring} \rightarrow \text{stereotype} : \text{SpringStereotype} \\
\text{stereotype} : \text{SpringStereotype} \rightarrow \text{ftr} : \text{jacket.loose} = 1, \\
\text{ftr} : \text{jacket.color.azur} = 1 ...
\]

Apart from the retrieval of a rigid stereotype, i.e., a stereotype that represents a style advice rule defined by a fashion expert, PServer offers the option of the retrieval of a flexible stereotype, i.e., a stereotype that corresponds to user preferences. Similar to the rigid stereotypes, a flexible stereotype is triggered when the appropriate attribute value is specified. In this manner two types of stereotype are selected: a stereotype corresponding to what experts consider appropriate and a stereotype corresponding to what other people consider appropriate.

- **Generating Recommendations.** The last step of the process is the generation of the recommendations. The recommendation engine is built as a client application to the PServer. The client collects the stereotype features and subsequently “consults” the ontology in order to select the appropriate garment which is represented as an individual having properties “similar” to the stereotype features. For example, the stereotype features of the example above are used to select a loose jacket with azur colour. This garment is delivered to the customer. Since we have two kinds of stereotype, rigid and flexible, the recommendation engine might recommend two different garments. In that case, the customer has the option to compare and finally “follow” the opinion either
of style experts or customers with similar preferences.

The above process can be seen as a “semantic” cycle which starts and ends with the ontology. A particular individual is initially mapped to the ontology, and is finally recommended a garment that fits her from the ontology. In this manner all the information of the fashion domain is stored in a common repository, described using a common language and exploited using a common interface.

7. Conclusions & Future work

In this work we presented a framework for delivering personalised style recommendations. To realize this task we created a fashion ontology that performs the role of a semantic repository for concepts and relations of all the available knowledge sources in the fashion domain.

Using the semantic information of this ontology, a set of style advice rules have been defined by fashion experts. These rules are stored into a general-purpose personalisation server, named PServer, by means of user stereotypes. These stereotypes are subsequently exploited to offer recommendation functionality to the users of the system.

The proposed methodology provides a promising research direction, where many new issues arise. The fashion ontology can be extended with new elements, order to cover more concepts from the fashion domain. Moreover, the style advice rules can be enhanced by experts in order to provide more product-specific recommendations.

There is also the option to exploit the PServer’s functionality more dynamically. PServer can use machine learning technologies to learn new style advice rules using data from user interactions with the system, or communities.

Another important issue is the presentation of recommendations to the end user. The authors have worked into enriching domain ontologies with linguistic structures, in order to generate natural language descriptions [12]. Let us consider a very simple example, provided the following individual (mary hasHighRec jacketFitted), and a relevant microplan (linguistic) annotation for the hasHighRec property then, the sentence A very good recommendation for mary is a fitted jacket will be produced.

Acknowledgement

This work was carried out in the context of project CP-TP 214455-2 SERVice Oriented Intelligent Value Adding nEtwork for Clothing-SMEs embarking in Mass-Customisation (SERVIVE), which is partially funded by the European Union.

References