Undefined 1 (2009) 1–5
IOS Press

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 this paper we present a knowledge framework that provides 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 a 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

In the fashion industry, mass-customisation is a new trend that tries to produce clothes respecting the idiosyncrasy of every customer and doing so cost effectively. Thus 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 industry. In particular, material properties, human morphology, garment styles, and occasion for a garment are among the domains that influence style advice. Although, there are associations between the above do-

*Corresponding author. E-mail: dimitrv@iit.demokritos.gr.

main (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, extracting the available knowledge and presenting 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 the personalisation process is a 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 system gathers the users' characteristics, which are a few body measurements and some facial features, including skin colour, and subsequently builds a user model. Based on this, the system assigns the user to a body type, as well as to a human colour style. Then the system suggests some generic garments to the current user, as well as garment colours pertinent to the colour style. The generic garments concern garments for the upper, the lower part, or the whole of the body. For instance, garments for the lower part could be skirts, dresses, and trousers. At a further level of specialisation, trousers could be distinguished into harem pants, or jeans. The suggestions of the system are positive, negative, or neutral. Negative is to be interpreted as "do not wear this ..."

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 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 related 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 knowledge representations in the fashion domain. BodyXML is an taxonomy based on XML that includes body and product (i.e. garment) representation [6]. BodyXML was conceived to tackle the issue of lack of standard representations for 3D and 2D body data in various fields of research and development (particularly in the area of garment design, manufacturing and retailing). Thus BodyXML primarily represents data from body scanners. 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

[1] www.servive.eu

multiple “representations — 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 has two attributes: “details” — unique information such as retailer and manufacturer names, textiles, care instructions etc; and multiple “representations” - 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 an ontology for garments [2]. This ontology includes an abstract description of the human body and an ontology for clothing patterns. For instance the human body is split into hulls such as the neck, arms and legs. On the other hand, a cloth pattern for a jacket would determine, the patterns that make up the jacket, as well as sewing information. Commonly used descriptions of garments include jackets, trousers, skirts and dresses. This ontology has been applied to modeling virtual clothing on avatars, as well as to the problem of cloth-cloth collision detection, which refers to representing the physical interaction between layers of clothing. Ontologies for Virtual Humans have been developed in [3]. Moreover, some ontologies for the human body have been developed in the health and biomedical industries [2].

The nowledge representation requirement of the current work, are somewhat different from the aforementioned taxonomies and ontologies. First, human bodies need to be described by rather high-level concepts, such as the body types, and possibly a small set of body measurements used to derive the body types, far from the needs in other domains where, for instance, a cloud of points is derived from a scanner. Second, the garments that form part of the recommender system, are described by features that are different from the patterns used in other ontologies. For instance, a military jacket in our ontology, is something that could be suggested to the end user, whereas the clothing patterns in other ontologies are aimed at a tailor. Moreover, our ontology also contains, information about groups of humans, as they are characterised by facial features that affect the garment colour suggestion. Finally, our ontology contains information about garment materials and their properties, which is not included in the above ontologies. As a result the SERVIVE ontology could complement some of the aforementioned ontologies in a future setting where a simulacrum of a prospective customer and his/her suggested garments is created.

[2]http://www.bioontology.org/projects/ontologies/fma/ fmaOwl-FullComponent_2_0.owl, last access Dec2010

Table 1
Example of a user model (atomic)

Attributes			**Features**	
user	*bodyType*	*gender*	*skirt.pleated*	*jacket.military*
Mary	Average	f	1	0.5

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 [11]. 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 [12]. 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 for women of average body type are pleated skirts are highly recommended, 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 though 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 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 stereo-

Table 2
Example of a stereotype (collective model)

Attributes	**Features**		
bodyType	*skirt.pleated*	*jacket.military*	*jacket.peplum*
Average	1	0.5	-1

Table 3
Example of a users' community (collective model)

skirt.pleated	*jacket*
0.9	0.8

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 is in the *Recommender systems (RS)*, which have evolved a lot and especially on the Web. RS is any system that produces personalised recommendations or guides to a user, towards interesting or useful items 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 favour.
– *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:

In *MyShoppingPal.com* a user submits information about his/her body type and style preferences. This information is used for suggesting garments. A similar system is the *MyShape* [3] recommendation system.

In *My Virtual Model* [4] 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 in a concept can be propagated to the super-concept, which can be integrated into the user's interest profile [9]. The above approach has been evaluated in the domain of recommending research articles, but it is obviously applicable to various ontologies.

2.1. Fashion recommender systems

Fashion recommender systems, build a user's profile and subsequently suggest clothing items that are appropriate for that profile, or even clothing items that can be combined. This is known as coordination. The major computational components that make up most fashion recommender systems include *rule based systems*, *content and collaborative based recommender systems* and *fuzzy neural networks*.

Rule-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 a rule based system for the mass customisation of garments, and in particular of shirts [8]. In that system, the users provide information about height, weight, complexion, as well as some subjective pieces of information. Then the user receives a suggestion, which is followed by 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 net-

[3] http://www.myshape.com/
[4] http://www.mvm.com/

work is employed [15]. Collaborative filtering has been used by the *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.

A combination of collaborative and content-based filtering is used by FDRAS [5] and *"what am I going to wear"*. In the latter system, each user creates his/her wardrobe that includes the garment categories according to his/her preferences. A recommender system either suggests 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.

The starting point of our approach is an OWL ontology, that represents the hierarchies, relations, and styling rules of the fashion domain. This in turn encodes fashion experts' knowledge and is roughly analogous to other rule-based systems with the exception of adopting a widely used ontology language. Then, parts of the ontology, and in particular the style advice rules are represented in a personalisation server (PServer) and they make up the stereotypes, upon which preliminary styling information can be passed on to the user. The PServer can maintain detailed user profiles based on their expressed preferences on garments. Subsequently, communities of users of similar preferences can be formed with PServer clustering algorithms. Moreover, dynamic stereotypes can be discovered based on usage data. Communities, and dynamic stereotypes can enhance the advice offered by the static stereotypes. Finally, dynamic stereotypes from the personalisation server can be inserted in the ontology to enrich it.

3. Knowledge Integration: Servive Fashion Ontology

Knowledge about garments, materials, various human styles, and human morphology are among the things pertinent in fashion advice. Although there are associations between the above types of information sources, 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 [4], with the aid of the Protégé 4.1 ontology editor. The ontology captures the experience of a 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 some pertinent *garment types* and *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 [5].

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, are 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 (18 subclasses)`. 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

[5]http://www.servive.eu/

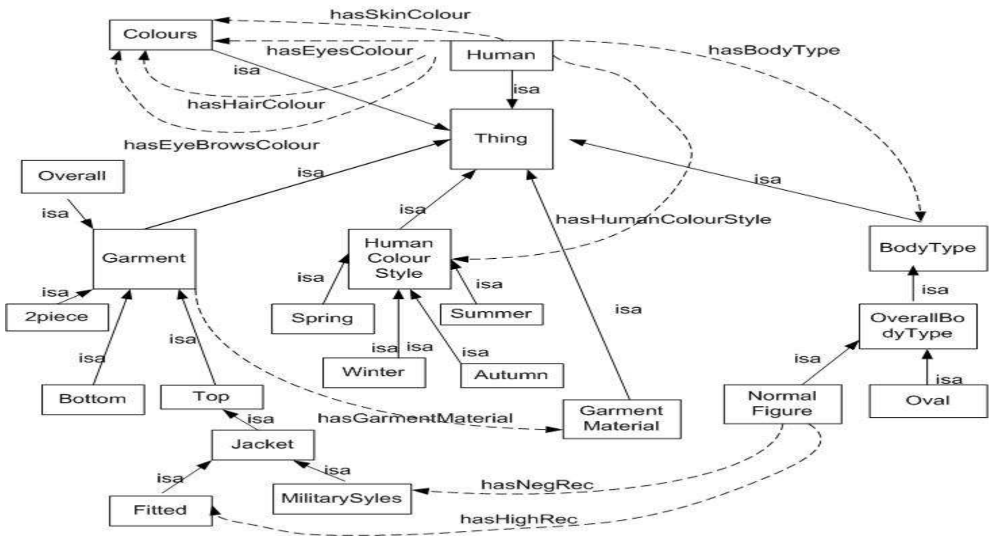

Fig. 1. Top level structure of the Servive Fashion Ontology. Solid lines denote `isa` relations, and dashed lines denote object properties

members (individuals). Examples of these individuals are the `navy, black, blue, grey`, etc.

– `class:Garment (8 subclasses)`. 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 (6 subclasses)`. The class models the characteristics that can be used to distinguish the various types of garment, such as Buttons, Cut, Pattern, etc. These characteristics correspond to the subclasses of the class.
– `class:GarmentMaterial (60 subclasses)`. 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:HumanStyleColour (4 subclasses)`. The class models the categories that a human can be assigned, based on the Season Analysis Model [7]. There are four different categories of human colour style that are represented as subclasses of `Human Style Colour` class, i.e., `Spring`, `Summer`, `Autumn` and `Winter`.
– `class:Occasion (7 subclasses)`. 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 (3 subclasses)`. 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, hasEyeBrows-Colour, 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 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. An issue that we faced was the representation of rules in OWL without recourse to SWRL rules. Next, we provide samples of those rules in a Prolog-like format as they were initially captured after contacting domain experts, as well as their final form in OWL.

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. They associate human facial features with "human style colours", i.e. summer, spring, autumn and winter. Such a rule, also known as a production rule would be stated as follows,

```
hasEyeBrowsColour (X,Y) and
hasEyeColour (X,Z) and
hasHairColour (X,W) and
hasSkinColour (X,N) and
(Y==light; Y=darkBlond; Y=Light) and
...
--> Spring(x)
```

In OWL the above can described as a defined class. Thus, if an individual (a customer in our case), provides input relevant to object properties that appear next, she will be classified to the Spring class by a reasoner supporting OWL 2, such as Pellet [13]. Similarly, there are three additional definitions for the Summer, Autumn and Winter classes.

```
Spring EquivalentTo
hasEyeBrowsColour some
   {Light, DarkBlond, GoldenNaturalBlond}
and hasEyeColour some
   {Aqua, Hazelnut, Green, Golden, LightBrown}
and hasHairColour some
   {Light, DarkBlond, GoldenNaturalBlond}
and hasSkinColour some
   {Light, Frekles, Golden}
```

Table 4
Number of Stereotypes or second level rules

Normal figure	39
Triangular	19
Hourglass	19
OvalRound	13
Narrow and Straight	17
Square	18
Summer	3
Winter	3
Spring	3
Autumn	3
total	137

Another set of attribute rules allows the specification of a body type based on body measurements. As a production rule, this would be stated as,

```
hasWaistHeight(X,Y), Y<=100.826 and
hasHeight(X,Z), Z>68 -->
OvalBodyType(X)
```

In OWL, the above can described as a defined class with the aid of data properties. Similarly, there are 7 more rules for the rest of the body types:

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

Note that the values of object and data properties are expressed in OWL as class descriptions with property restrictions.

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. Statistics about the style advice rules are provided in table 4. In a production system a rule such as the following captures the advice,

```
normalFigure(X) -->
    hasHighRecom (X, jacketFitted)
```

In OWL, we represent the style advice rules also as class definitions. For above rule, and 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
```

The definition of individuals, representing garments, was a technicality necessary for the operation of a reasoner. Given the above definition, and `mary` 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.

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". [6] It has been used for personalisation in a variety of fields [10]. 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 modeling 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 exploiting user interactions with the system and in particular, frequency counts and/or histories of actions in order to update the feature values of personal user models and user stereotypes. In this manner, it is possible to infer the level of interestingness of each user in a certain feature.

[6] http://www.iit.demokritos.gr/skel/

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 by means of stereotypes. To realise this, a mapping is required from the elements of SFO, i.e., classes, subclasses, etc to the elements of PServer, i.e., attributes and features. This is described next with concrete examples.

To start with, SFO classes are mapped to PServer attributes. These classes refer to human body characteristics, such as body type, age or facial features, or attribute rules,

$$class : BodyType \rightarrow attr : BodyType$$

Classes, can also be mapped to PServer attribute values,

$$subclass : NormalFigure \rightarrow$$
$$attr\{BodyType\}.Value : NormalFigure$$

$$subclass : Spring \rightarrow$$
$$attr\{HumanColourStyle\}.Value : Spring$$

Finally, SFO classes can be mapped to PServer features,

$$class.subclass : Jacket.Military \rightarrow ftr : Jacket.Military$$

Another element of the SFO that is mapped to the PServer are the object properties *hasHighRec*, *hasLowRec* and *hasNegRec* that associate humans to garment with high, low, or negative recommendedations. Each object property is associated with a *feature value* in the Pserver, as in the example below:

Table 5

Example of a stereotype (with SFO elements)

Attributes	**Features**		
class{BodyType}	*class{Garment}.subclassOf{GarmentFtrs}*	*class{Garment}.subclassOf{GarmentFtr}*	*class{Garment}.subclassOf{GarmentFtr}*
subclass{BodyType}	objProp{hasHighRec}	objProp{hasLowRec}	objProp{hasNegRec}

$$objProp : hasHighRec \rightarrow ftr\{Jacket.Military\}.Value = 1$$

$$objProp : hasLowRec \rightarrow ftr\{Jacket.Military\}.Value = 0$$

$$objProp : hasNegRec \rightarrow ftr\{Jacket.Military\}.Value = -1$$

Following the above rationale, a presentation of the stereotype of Table 2 with the SFO elements is given in Table 5. Thus, given the following SFO construct, expressing a second level style advice rule,

```
Oval EquivalentTo hasHighRec value jacketLoose
```

it will be mapped to the following PServer's stereotype,

```
attr:BodyType=Oval, ftr:jacket.loose=1
```

These stereotypes can subsequently be exploited to deliver recommendations. Since they have been created manually and are based on second level style advice rules, they are considered as *recommendations from fashion experts*.

There are a number of advantages of this approach, compared to fashion recommender systems mentioned in the literature review. In particular:

– *Stereotype confirmation/rejection.* PServer allows the statistical analysis of user stereotypes based on their usage. In other words, each time a client application *triggers* a stereotype from the PServer, this action is recorded and the relevant information is updated. For example, given the stereotype: "Oval women are advised to wear pleated skirts", we might discover from user transactions, recorded in a client application, that the "Oval people" actually buy peplum skirts. This discovery corresponds to a disapproval of the style advice rule, and thus it could subsequently be removed from the set of stereotypes.

– *PServer & Ontology Enrichment.* The discovery from user transactions of new stereotypes, might enrich the system, which is realised by either creating a new stereotype in the PServer, which can be mapped as a new style advice rule in the SFO, for example:

```
Oval hasStatisticalRecom some {skirtPeplum}
```

where *hasStatisticalRecom* is an object property of the SFO, denoting the origin of the recommendation.

– *Easier access to stereotype information.* By mapping the semantic information to user stereotypes it creates a more personalised view of the knowledge that is available in the fashion domain. This personalised view can be managed and retrieved by external applications using the PServer's interfaces a lot easier than accessing directly the SFO structure. In addition, the hierarchical structure of features in PServer, allows the client application to retrieve stereotypes that cover a broader range of garment features, for example *Jacket.**.

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

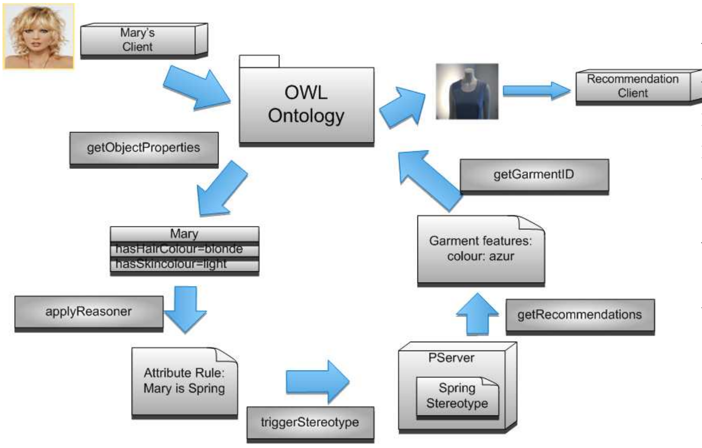

Fig. 2. Ontology & PServer Integration.

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 blond. 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=blond".
– *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 & Update*. Having inferred the individual's attributes, the PServer's stereotypes are triggered. This step is realised 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,

$attr.rule : Spring \rightarrow attr : Spring\}$

$attr : Spring \rightarrow stereotype : SpringStereotype$

$stereotype : SpringStereotype \rightarrow$
$\quad ftr : jacket.loose = 1,$
$\ldots$
$\quad ftr : jacket.color.azur = 1$

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 updating the statistical information of stereotype's usage and thus, either confirm the "importance" of the stereotype or its rejection by users.
– *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.

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 realise 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, in order to cover more concepts from the fashion domain. In the near future, the ontology will be extended to include style advice rules that incorporate the occassion for a garment suggestion.

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. Finally, the newly discovered style advice rules can be mapped back to the SFO in order to enrich it.

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 [14]. 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

[1] G. Adomavicious and A. Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[2] A. Fuhrmann, C. Gross, and A. Weber. Ontologies for virtual garments. In *Proceedings of Workshop towards Semantic Virtual Environments*, 2005.

[3] A. Garcia-rojas, D. Thalmann, F. Vexo, L. Moccozet, N. Magnenat-thalmann, M. Mortara, and M. Spagnuolo. An ontology of virtual humans: incorporating semantics into human shapes. In *EWIMT*, pages 7–14, 2005.

[4] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.

[5] K.Y. Jung, Y.J. Na, D.H. Park, and Jung-Hyun Lee. Discovery knowledge of user preferences: Ontologies in fashion design recommender agent system. In *Computational Science and Its Applications - ICCSA 2004*, pages 863–872. Springer, 2004.

[6] G. Kartsounis, I. Douros, and P.C. Treleaven. BodyXML - An XML-based High Level Representation Framework of Human Bodies. In *Proceedings 3D Modelling, Paris*, 2003.

[7] Bernice Kentner. *Color Me a Season: A Complete Guide to Finding Your Best Colors and How to Use Them*. Ken Kra Publishers, 1979.

[8] N. Liu, T.M. Choi, C.W. Yuen, and F. NG. An exploratory study on expert system's applications in fashion mass customisation. In *Proceedings of the Eighth Conference on Machine Learning and Cybernetics, Baoding, China*, 2009.

[9] S. E. Middleton, N. R. Shadbolt, and D. C. De Roure. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems*, 22(1):54–88, 2004.

[10] G. Paliouras, A. Mouzakidis, V. Moustakas, and C. Skourlas. PNS: A Personalized News Aggregator on the Web. In M. Virvou and L. Jain, editors, *Intelligent Interactive Systems in Knowledge-Based Environments*, volume 104, pages 175–197. Springer, 2008.

[11] D. Pierrakos, G. Paliouras, C. Papatheodorou, and C.D. Spyropoulos. Web usage mining as a tool for personalization: a survey. *User Modeling and User-Adapted Interaction*, 13(4):311–372, 2003.

[12] E. Rich. User modeling via stereotypes. *Cognitive Science*, 4:329–354, 1979.

[13] E. Sirin, B. Parsia, B.C. Grau, A. Kalyanpur, and Y. Katza. Pellet: A practical owl-dl reasoner. *Web Semantics*, 5(2):51–53, 2007.

[14] D. Vogiatzis, D. Galanis, V. Karkaletsis, I. Androutsopoulos, and C.D. Spyropoulos. A conversant robotic guide to art collections. In *2nd Workshop on Language Technology for Cultural Heritage Data, Language Resources and Evaluation Conference (LREC 2008)*, 2008.

[15] W.K. Wong, X.H. Zeng, and W.M.R. Au. A decision support tool for apparel coordination through integrating the knowledge-based attribute evaluation expert system and the T-S fuzzy neural network. *Expert Systems with Applications*, 36:2377–2390, 2009.