A casting company has a large database that consists of person-models. Advertisement companies are using this database to look for models to be used in advertisements or other activities. Each entry in the database contains a photo of the model, personal information and some body and face characteristics. The casting company has created a user interface for inserting the information of the models as instances of a predefined ontology. It also provides a query engine to search for models with specific characteristics. A user can query the database providing high-level information about the models (such as the name, the height, the type of the hair etc.).

We will provide some examples in order to highlight the need of fuzzy extension in the ontology, rule and query languages. Let us consider that an advertisement company requires a female model who is tall and thin. This query is formed as:

Query("List all the female models, which are
over 175 cm and under 60 kilos")

The query pattern is as follows:

Query Pattern {(hasSex ?p ?a)(type ?a female)(hasHeight ?p ?c)
(type ?c Height $\geq$ 175)(hasWeight ?p ?d)(type ?d Weight $\leq$ 60)}
Answer1: ("Mary is a female model who is over 175 cm and is under 60 kilos")
Answer2: ("Susan is a female model who is over 175 cm and is under 60 kilos")

In a 700 models database the answers that make the query true (entails the KB) are "Mary and Susan". However, after a closer look in the database, we find out that there are more than 50 models that could satisfy *to some degree* this query if we didn't have crisp thresholds. In conjunctive queries, if one of the atoms of the query does not entails the KB we get an empty answer. If, for example, the model "Adriana", which satisfies the thin sentence, but is under 1cm only in the height sentence, is rejected.

Another type of uncertainty is introduced when the query sentences are implemented with an equal degree of importance. For example, the advertisement company could be more interested, for the model, to be tall than to be thin. This means, apart from limited query answers, that we cannot rank the answers of the query according to the user needs. If, for example, "Mary" is 185cm tall and 65 kilos and "Susan" is 185cm tall and 55 kilos and the degrees of importance of the atoms is 1 and 0.5 for the weight and the height respectively, then "Susan" should be ranked before "Mary".

The above problems can be solved if we use a fuzzy KB (fuzzy Abox) to retrieve the answers, which means that we can use fuzzy concepts and fuzzy relations in the queries, such as "tallPerson, fatPerson, hasMediumHeight" together with assertion degrees representing the membership value of the object to the corresponding concepts and relations sets. In addition to the fuzzy assertion, the user may assign degrees of importance to the query sentences denoting the influence that a specific sentence must have in the query answer. For example, in the query *"List all the models that hasLargeHeight and hasMidleAge"* the user might be more interested in the height sentence than in the age sentence. In this case, the user can assign degree of importance 1 to the height sentence and 0.5

to the age sentence. The example presented in the previous section has the form in the fuzzy case:

Query("List all the female models, which are
tall and thin")

The query pattern is as follows:

Query Pattern {(hasSex ?p ?a)(type ?a female)∧(hasLargeHeight ?p ?c <0.8>)
(type ?c LargeHeight)∧(hasMediumWeight ?p ?d <0.5>)(type ?d MediumWeight)}
Must-Bind Variables List: (?p)
Answer1: ("Mary is a female model who is 185cm tall (largeHeight=0.8)
and is 65 kilos (mediumWeight=0.4)
Answer2: ("Susan is a female model who is 175cm (LargeHeight=0.6)
and is 50 kilos (mediumWeight=0.9)
Answer3: ("Helen is a female model who is 170cm (LargeHeight=0.5)
and is 50 kilos (mediumWeight=0.9)

In this example we used the fuzzy relations "hasLargeHeight, hasMediumWeight" and the fuzzy concepts "LargeHeight, MediumWeight" to manage the uncertainty introduced by the concepts "Thin, Tall". In this way, a model who is 183cm tall hasLargeHeight=0.65 and hasMediumHeight=0.3, as depicted in figure 1. Also we have assigned degrees of importance, 0.8 for the height sentence and 0.5 for the weight sentence. That is, the user is more interested for the model to be tall that to be thin.

In the following paragraphs we provide a sample of the Tbox, the Abox, two inference rules rules and a diagram showing how the assertion degrees are calculated from the textual information of the models.

<u>Tbox</u>

$Woman \equiv Person \sqcap Female$
$Man \equiv Person \sqcap Male$
$CastingPerson \equiv Person \sqcap \forall HasPersonalInformation.PersonalInformation$
$\sqcap \forall HasMeasurements.Measurements \sqcap \forall HasTypeOfHair.Hair$
$PersonalInformation \equiv \forall hasName.Name \sqcap \forall hasLastName.Name \sqcap$
$\forall hasAge.Age$
$\sqcap \forall hasDOB.DOB \sqcap \forall hasAddress.Adress \sqcap \forall hasMobilenumber.Number$
$Measurements \equiv \forall hasHeight \sqcap \forall hasWeight.Weight$
$\sqcap \forall hasShoeSize.Size$
$Hair \equiv \exists hasHairQuality.HairQuality \sqcap$
$\exists hasHairLength.HairLength \sqcap \exists hasHairColour.HairColour$

Entry: no1		
Personal Information		
Name: Vassilis	*LastName:* Tzouvaras	*Age:* 29
Address: Hatzi 7	*Mobile:* 6937295722	*D.O.B.:* 07.08.75
Measurements		
Height: 183cm	*Weight:*90	*ShoeSize:* 44
Hair		
Quality: good	*Length:* short	*Style:* frizy

<u>**Abox:**</u>

$\{\langle no1 : CastingPerson = 1\rangle,\ \langle (no1, Vassilis) : hasName = 1\rangle,\ \langle (no1, Tzouvaras) : hasLastName = 1\rangle,\ \langle (no1, 29) : hasAge = 1\rangle,\ \langle (no1, Hatzi7) : hasAddress = 1\rangle,\ \langle (no1, 6937295722) : hasMobilenumber = 1\rangle,\ \langle (no1, 183cm) : hasMediumHeight = .3\rangle,\ \langle (no1, 183) : haslargeHeight = .65\rangle,\ \langle (no1, 34) : mediumWaste = 0.7\rangle,\ \langle (no1, 34) : hasLargeWaste = 0.3\rangle,\ \langle (no1, 44) : hasMediumShoeSize = .9\rangle,\ \langle (no1, 44 : hasLargeShoeSize = 0.1\rangle,\ \langle (no1, Long) : hasLongHair = 0.3\rangle,\ \langle (no1, good) : hasQualityHair = 0.8\rangle,\ \langle (no1, frizy) : hasTypeOfHair = 1\}$

Rule 1: $IF\quad hasMediumWeight(a, c)\quad AND\quad hasLargeHeight(a, b)\quad THEN\quad ThinPerson(a)$
Rule 2: $If\quad HasSmallHeight(a, c)\quad AND\quad HasLargeWeight(a, b)\quad THEN\quad FatPerson(a)$

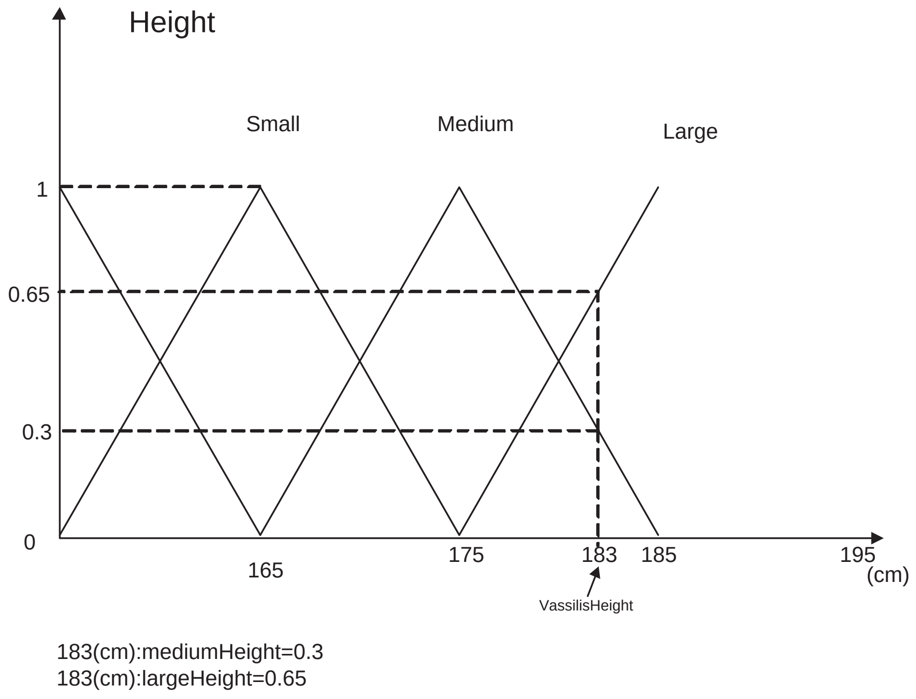

Fig. 1. The fuzzy partition of Height