INFORMATION TECHNOLOGIES FOR SHIFT TO RAIL

D5.7 – Travel Companion Ontology

Due date of deliverable: 31/10/2017

Actual submission date: 20/11/2017

Leader of this Deliverable: Polimi

Reviewed: Y

Document status		
Revision	Date	Description
1	24/7/2017	Initial version
2	31/8/2017	After comments
3	20/11/2017	Final version

Project funded from the European Union's Horizon 2020 research and innovation programme		
Dissemination Level		
PU	Public	X
CO	Confidential, restricted under conditions set out in Model Grant Agreement	
CI	Classified, information as referred to in Commission Decision 2001/844/EC	

Start date of project: 01/05/2015 Duration: 36 months

EXECUTIVE SUMMARY

This document presents the part of the IT²RAIL ontology that concerns the Travel Companion. After a brief introduction, the document presents the modelling choices that were taken in the ontology design with particular focus on preference modelling. Then, the ontology concepts are explained in detail, including some constraints that define their semantics.

This deliverable extends and revises Deliverable D5.1.

TABLE OF CONTENTS

LIST OF FIGURES

1 LIST OF ABBREVIATIONS

ACK	Acknowledge
BA	Business Analytics
BT	Booking & Ticketing
CDT	Context Dimension Tree
CRUD	Create, Read, Update, Delete
CW	Cloud Wallet
E-R	Entity Relationship
GUI	Graphical User Interface
NACK	Not Acknowledge
PA	Personal Application
TC	Travel Companion
TS	Travel Shopper
TT	Trip Tracker
UI	User Interface
UUID	Universally Unique IDentifier

2 INTRODUCTION

This document presents the fragment of the IT²RAIL ontology that concerns Travel Companion (TC). The document focuses on the concepts that are specific to TC, that is, the user preferences. Other concepts, such as journeys, bookings, etc. that are relevant for TC, but which are within the purview of other IT²RAIL functional areas, are not included in this document; their description can be found in ontology deliverables those areas. Moreover, the concepts mentioned in this document are those that are part of the interfaces, as described in [1]; hence, the aim of this document is not to define a data model for TC, but only to provide a description of the concepts that are in the interfaces.

The main results exported by the TC module are constituted by the preferences; therefore, the TC ontology deals with defining what a preference is and how it is used by the TC.

This deliverable is both an extension and a revision of its first iteration [2]. In particular, some concepts have been added. Also, four categories of preferences have been identified in Section 5. Those four categories are characterised by specific semantic constraints, which have been newly introduced in the current deliverable.

The document describes in Section 3 the methodology chosen to define and model preferences. Section 4 provides a description of the concepts included in the ontology and their relationships. Section 5 separates preferences in different categories, and for each of them it introduces some constraints defining their semantics.

3 METHODOLOGY

Before presenting the list of preferences in Section 4, this section summarises the general modelling choices — independent of the specific preferences — adopted by the TC to represent preferences; a more exhaustive discussion about this topic can be found in the TC specification document [1].

When organising a travel, a traveller can choose her travel features from a general list of options. From this list we can express different kinds of preferences, e.g. Jane is Vegetarian (this is a “stable” preference that we do not expect to change often), she is going on a working trip (this is an indication of a context in which the travel occurs and it might influence the choice of other preferences) and she wants to travel with cabin luggage only (this might be a preference that is selected by the traveller on a per-travel instance).

The preference model considered in IT²RAIL has been defined on the basis of the CDT model [1], thus it allows us to represent contextual preferences. Every preference is associated with a score which can assume different semantics: (i) a preference can be just a binary choice (YES/NO) or (ii) a preference can be expressed as a numerical score in some interval, such as [0, 1], or [-1, 1].

A contextual preference can be expressed as a quadruple <*userID, context, SQ, score*>. This quadruple expresses the fact that the user (identified by *userID*), in a certain *context*, prefers the items indicated by the “selection query” *SQ* with the given *score* (where the “selection query” *SQ* is simply a condition that selects a preference, such as for example “class = First”, which selects all items — e.g., offers — whose attribute “class” is “First”). So, for example, if Jane is travelling alone she prefers to be seated near the window rather than near the aisle, similarly if she travels with friends. However, while she was pregnant she realised that the aisle is much better if one has to get up a lot, therefore, during a pregnancy, and also when travelling with children, she likes it better to seat near the aisle. Moreover, Jane knows that in business class seats are much larger and food is better than in economy. Thus, when travelling in business class, she really likes to eat meat and sit near the window.

According to this discussion the TC models a collection of preferences each related to a certain context and characterised by its rank. The TC handles a specific set of preferences derived from those sent by the partners of the project.

To model these requirements, we studied two different ways of representing the preferences in Capella. The first possibility is to represent a preference as a class characterised by a name and a value, each one of them as strings. Then, an instance of this class — i.e., a specific preference — would be, for example, the pair name=“Preferred means of transportation” and value=“train”. This modelling is very compact, however, the TC handles a specific set of preferences and each of them has a domain that might differ from the others and this modelling choice does not allow us to specify explicitly the preferences handled by the TC, nor their domain.

For these reasons, we decided to study an alternative representation where a preference is still a Capella class, but it is further specialised, through a hierarchy, in a set of classes, each identifying a specific preference. Thus, the preference *Preferred means of transportation* is a class that is a specialisation of the class *Preference*. Moreover, the value of each preference is represented in

terms of an attribute whose domain is specified using an enumeration that allows us to list all the possible values the preference can assume.

4 DESCRIPTION OF THE CONCEPTS

The following Capella class diagram represents the concepts of the TC ontology and their relationships. A description of the employed concepts follows.

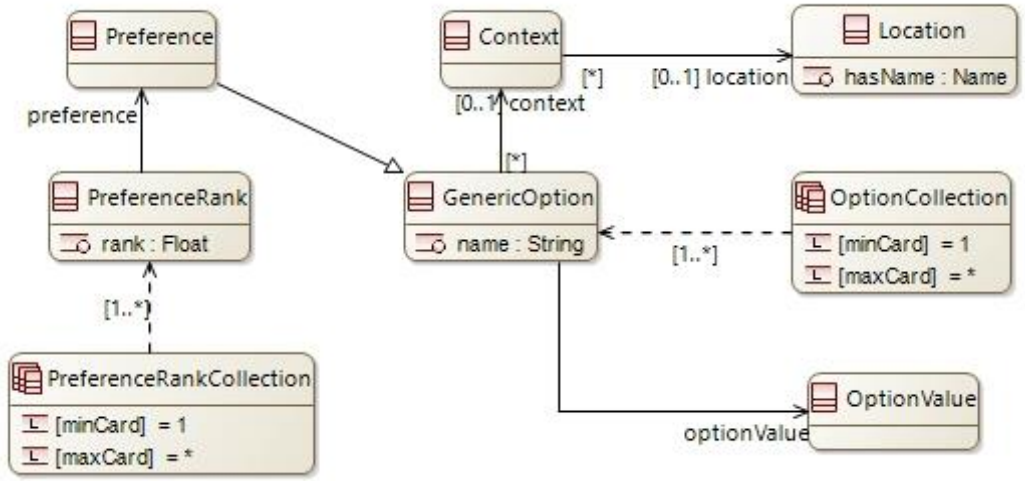

Figure 1 Capella class diagram representing the concepts of the TC ontology

First of all, a general notion of "option" (*GenericOption*) is introduced, to allow for a greater flexibility in defining not only "preferences", but also possibly similar concepts such as application options (which, unlike preferences, are not intended to be transmitted to other components of the system). The TC works with a collection of generic options (*OptionCollection* in Figure 1) such that one or more options might be defined. Each *GenericOption* is characterised by its name and assumes a certain value (as is described further in the document). As an example, our user Jane might like to travel by train, thus Jane's "Preferred means of transportation" assumes value "Train". Moreover, as described in the specification document [1], options (and in particular preferences) might be contextual, thus, they are related to a certain *Context* which is in turn related to a specific *Location*. For example, Jane might like to travel by train only when she is travelling for leisure.

As mentioned above, *Preferences* are a specialisation of *GenericOptions*. In general, a rank can be assigned to each preference, stating "how much" such preference is desirable. Assuming that the rank belongs to an interval [0, 1], imagine that, for example, Jane likes to travel both by train and by boat, however she prefers the train better; thus, the two preferences can be assigned a rank, e.g., "Preferred means of transportation = train" has rank 0.7 while "Preferred means of transportation = boat" has rank 0.4. *PreferenceRank* is the concept that associates a rank (a floating number in some interval) with a specific preference; in addition, since the TC deals not only with single preferences (and their ranks), but also with collections thereof, the notion of *PreferenceRankCollection* is introduced.

The Capella class diagram shown in Figure 2 and Figure 3 (which has been split for readability reasons) describes in detail the set of preferences the TC works with.

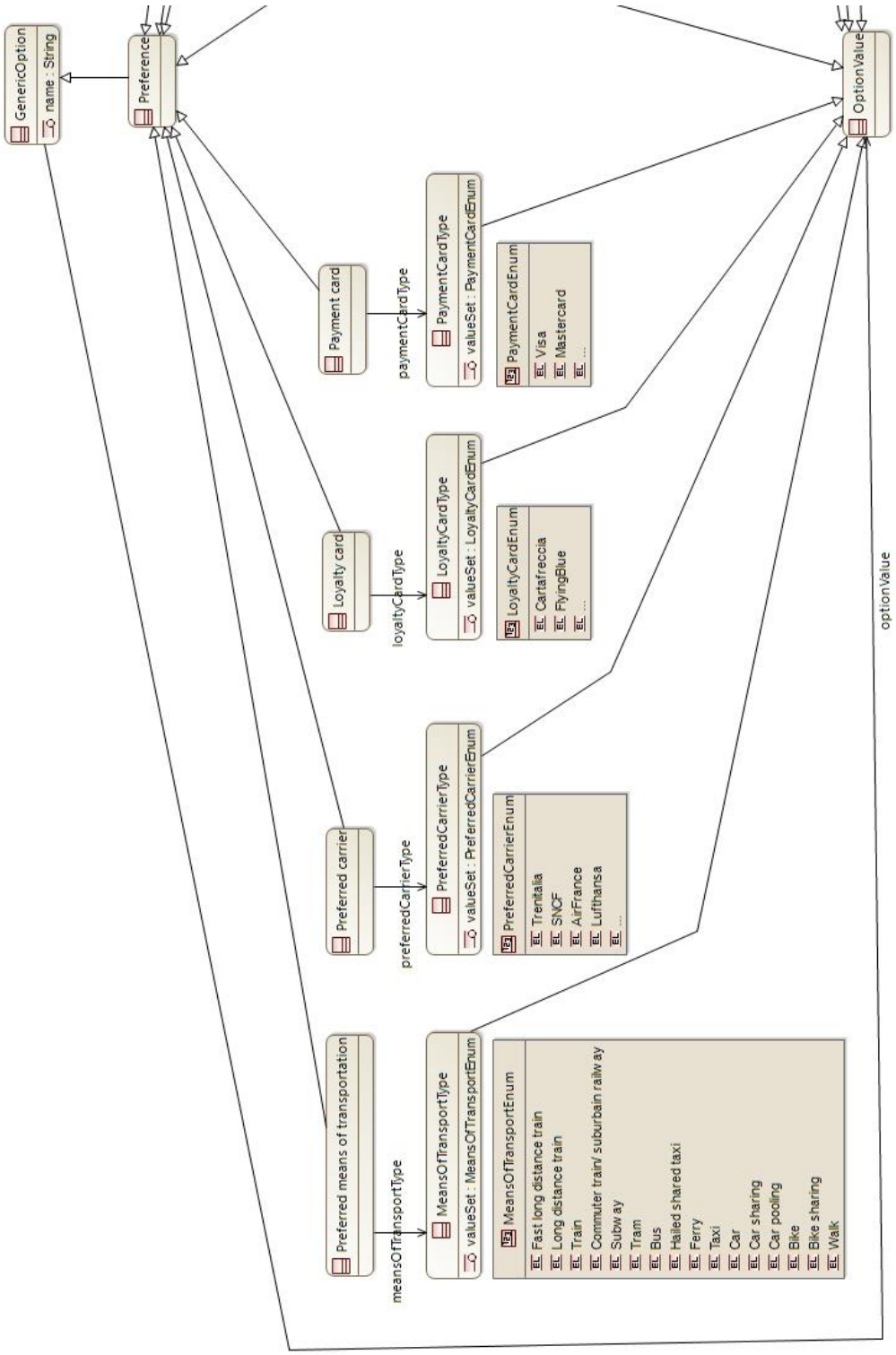

Figure 2 Capella class diagram representing the details of the Preference class (part 1).

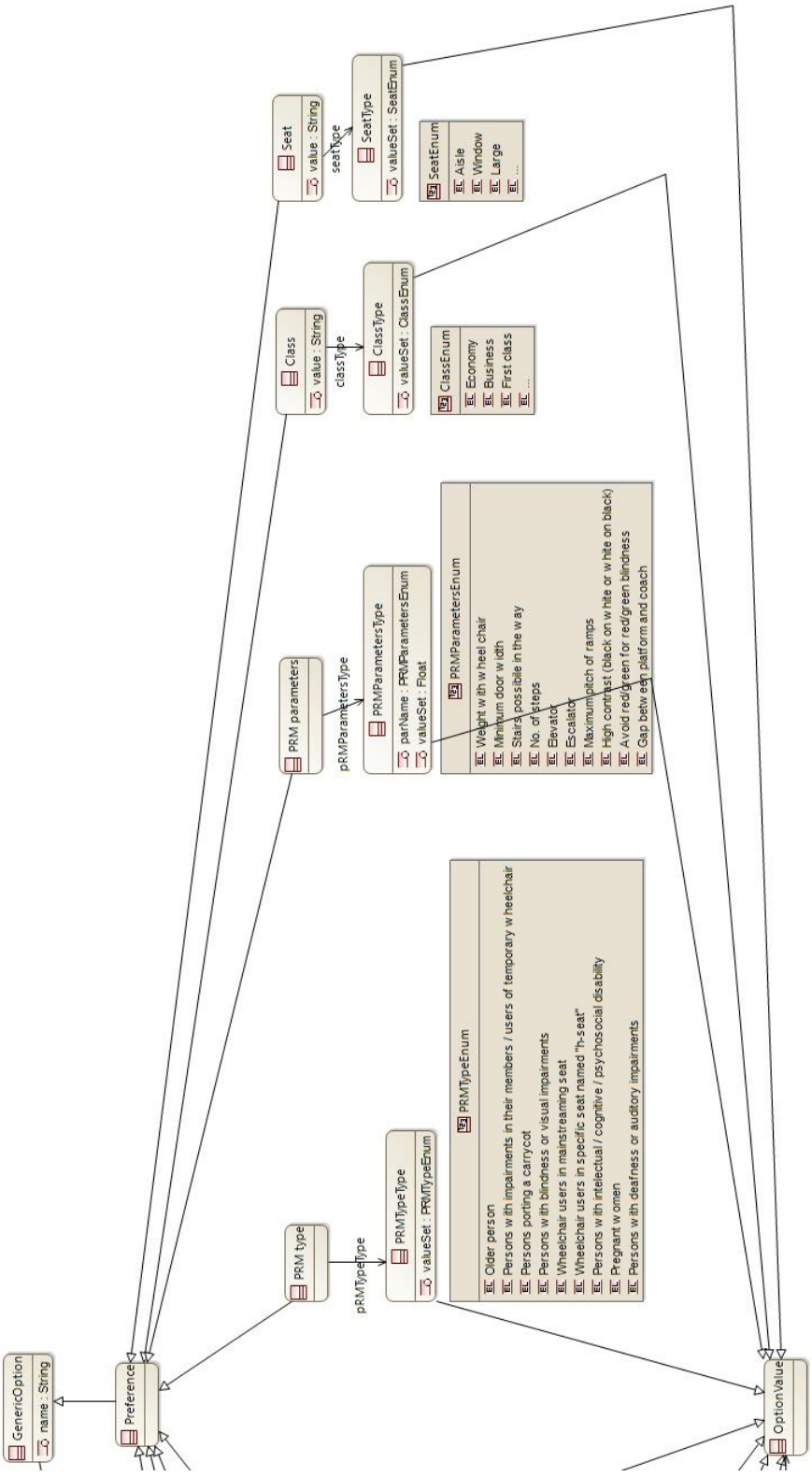

Figure 3 Capella class diagram representing the details of the Preference class (part 2).

As already mentioned, each preference is assigned a value whose domain strictly depends on the preference. The table in Table 1 summarises the preference types and the associated values.

Field Name	Field value	Journey Preference	Profile Preference	Trip Tracking Preference
Preferred means of transportation	• Train • Urban • Coach • Airline	Yes		
Preferred carrier	• Trenitalia • SNCF • AirFrance • Lufthansa	Yes		
Loyalty card	• Cartafreccia • FlyingBlue		Yes	
Payment card	• Mastercard • Visa		Yes	
PRM type	• Older person • Persons with impairments in their members / users of temporary wheelchair • Persons porting a carrycots • Persons with blindness or visual impairments • Wheelchair users in mainstreaming seat • Wheelchair users in specific seat named “h-seat” • Persons with intellectual / cognitive / psychosocial disability • Pregnant women • Persons with deafness or auditory impairments		Yes	
PRM parameters	• Weight with wheel chair • Minimum door width • Stairs possible in the way		Yes	

	• No. of steps • Elevator • Escalator • Maximum pitch of ramps • High contrast (black on white or white on black) • Avoid red/green for red/green blindness • Gap between platform and coach			
Class	• Economy • Business • First class	Yes		
Seat	• Aisle • Window • Large	Yes		
Trip Tracker behavior	• Automatic tracking activation • Offer alternatives			Yes
Message type	• Information • Warning			Yes
Message content	• Cancellation message • Rerouting message • Platform change • No first class • No dining car • No refreshment • WC out of order • Air conditioning / heating our of order • Wi-Fi inaccessible • Newspapers and magazines not available			Yes
Delays Parameters	• Significant delay • Absolute connection time • Marginal connection time • Avoid message duplication • Minimum delay change			Yes

Table 1 Preference name and value domain

Table 1 indicates, for each preference, whether it is a “profile” preference (i.e., if it changes very unfrequently, if at all), a “journey” preference (i.e., it is related to journey selection), or a “Trip Tracking” preference (i.e., it is used to determine the kind of alerts to be sent to the user). This grouping of preferences is mostly for convenience’s sake, and it does not necessarily signal a different semantics among the various types of preferences.

The next chapter presents in some detail the semantics associated with preferences, and in particular what ranks can be associated with different categories of preferences, and how they influence the behaviour of the application.

5 ON THE SEMANTICS OF PREFERENCES

As mentioned in [1], preferences are typically (though not exclusively) used during the shopping process, for filtering undesired offers, and/or to rank offers according to their suitability to the user. In general, preferences have the following features:

- Each preference (name, value) is assigned a score between a *min* and a *max*, where *min* and *max* can be arbitrary (for example, [0, 1], or [-1, 1]).
- A preference having the *max* score represents a **mandatory requirement** (i.e., it is used to filter the results).
- A preference having the *min* score represents a **mandatory exclusion** (i.e., it is used to filter the results).
- Any score between *min* and *max* is used for ordering the results of queries (typically involving itinerary offers). In particular, the score in the middle of the interval ((*min*+*max*)/2) means "don't care".
- Each result of a query (typically an offer) is given a score obtained as the composition of the single scores of each preference.

We categorise the **preferences on offers** into 4 categories. For each category, some constraints are associated with the preferences of that category. Each category of preferences can include multiple types of preferences, as detailed in Section 5.1.

- **Category 1**: a travel episode[1] can satisfy **just one travel mode** of the preference values at a time (e.g., a travel episode may be performed either by train or coach, but not both); more than one value with *min* score is allowed, while the *max* score is not allowed.
- **Category 2**: this category essentially includes the "Payment card" preference; in this case, a travel episode can satisfy **more than one** of the preference values at a time (e.g., a travel episode may allow travellers to pay only by Visa, only by Mastercard, or both); a *min* score is not allowed, but a *max* score is.
- **Category 3**: this category essentially includes the "Loyalty card" preference; a travel episode can satisfy **more than one** of the preference values at a time (e.g., a travel episode may allow travellers to use only Millemiglia, only FlyingBlue, or both); neither *min*, nor *max* scores are allowed.
- **Category 4**: preferences representing mandatory requirements: only the max score is allowed and more than one max score is allowed on the same preference (e.g., the same person can have more than one kind of PRM preferences).

Each resulting offer is given a score obtained as the composition of the single scores of each preference. In addition to preferences on offers, TC also handles **Trip Tracker preferences** (e.g., preferences concerning notifications about cancellations, rerouting, etc.), which are used by the traveller after a travel purchase.

[1] Recall that, in the IT²RAIL ontology, a "travel episode" corresponds to a single leg of a journey, which can have multiple legs. As such, a travel episode is characterised by a single mode of transportation.

In the rest of this chapter, first some further details concerning the semantics of the 4 categories introduced above are presented. Then, a brief explanation is given on how the scores are used to filter results of queries, and to rank them after filtering. Finally, some considerations are introduced concerning Trip Tracker (TT) preferences.

5.1 SEMANTICS OF PREFERENCES ACCORDING TO THEIR CATEGORY

In the following, we detail the semantics of each preference category.

Category 1

- Applied to a single travel episode.
- A travel episode can satisfy **just one travel mode** of the preference values at a time (e.g., a travel episode may be performed either by train or coach, not both).
- Allowed scores are in the [*min*, *max*) interval, thus *min* score is allowed and *max* score is not, meaning that it is a "limit" value, but it is never reached.
- More than one *min* score is allowed on the same preference, e.g.:

  score(User=Jane, Preference=PMT, Value=Train) = *min*

  score(User=Jane, Preference=PMT, Value=Coach) = *min*

- No travel episodes satisfying a preference with *min* score should be retrieved.

The preferences belonging to Category 1 are:

- Preferred means of transportation
- Preferred carrier
- Class
- Seat

The following logic formulae capture the semantics of the "Preferred means of transportation" preference. More precisely, we first define that, if a "Preferred means of transportation" has score *min*, then it cannot be present in the corresponding *ItineraryOffer*:

$\forall$ itOff : ItineraryOffer, pfRank : PreferenceRank, pfMeanTr : PreferredMeansOfTransportation,
tMean : MeansOfTransportEnum, itin : Itinerary, jny : Journey, trEp : TravelEpisode
(isBuiltWith(itOff,pfRank) $\wedge$
hasPreference(pfRank)= pfMeanTr $\wedge$ hasValueSet(hasOptionValue(pfMeanTr)) = tMean
$\wedge$
hasRank(pfRank) = min $\wedge$ hasItineraryOffer(itin,itOff) $\wedge$ hasJourney(itin,jny) $\wedge$
hasTravelEpisode(jny,trEp)

⇒

¬hasMode(trEp,tMean))

The next formula states that there cannot be a “Preferred means of transportation” with value *max*:

∀ pfRank : PreferenceRank, pfMeanTr : PreferredMeansOfTransportation,
tMean : MeansOfTransportEnum
(hasPreference(pfRank)= pfMeanTr ∧ hasValueSet(hasOptionValue(pfMeanTr)) = tMean
⇒ hasRank(pfRank) < max)

Similar formulae hold for the other types of preferences of Category. To exemplify this, we present the formulae for preference “Preferred carrier”.

∀ itOff : ItineraryOffer, pfRank : PreferenceRank, prCarr : PreferredCarrier,
carr : PreferredCarrierEnum, itin : Itinerary, jny : Journey, trEp : TravelEpisode
(isBuiltWith(itOff,pfRank) ∧
hasPreference(pfRank)=prCarr ∧ hasValueSet(hasOptionValue(prCarr)) = carr ∧
hasRank(pfRank) = min ∧ hasItineraryOffer(itin,itOff) ∧ hasJourney(itin,jny) ∧
hasTravelEpisode(jny,trEp)
⇒
¬hasCarrier(trEp,carr))

∀ pfRank : PreferenceRank, prCarr : PreferredCarrier, c : PreferredCarrierEnum
(hasPreference(pfRank)=prCarr ∧ hasValueSet(hasOptionValue(prCarr)) = carr
⇒
hasRank(pfRank) < max)

Category 2: Payment Card

- Some travel episodes might not allow to pay online (e.g., local transportation).
- Applied to a single travel episode.
- A travel episode can satisfy **more than one** of the preference values at a time (e.g., a travel episode may allow to pay only by Visa, only by Mastercard, or both).
- If a travel episode satisfies more than one preference value at a time, only the value with the greatest score is considered to compute the overall score of the travel episode.

- Scores must be in the (*min*, *max*] interval, thus *min* score is not allowed and *max* scores is. The *min* score is excluded because it does not seem sensible to filter out offers allowing the use of a certain card. Conversely, the *max* score is allowed, and it is used to say that, when the online payment is allowed, one of the cards with the *max* score must be allowed.

The following logic formulae capture the semantics of the “Payment card” preference. More precisely, if a “Payment card” has score *max*, then there must be at least one *TravelEpisode* in the offer that can be paid using that card:

$$\forall \text{ itOff : ItineraryOffer, pfRank : PreferenceRank, payCard : PaymentCard, card : PaymentCardEnum, itOffItm : ItineraryOfferItem}$$
$$(\text{isBuiltWith(itOff,pfRank)} \wedge \text{hasPreference(pfRank)=payCard} \wedge \text{hasValueSet(hasOptionValue(payCard))} = \text{card} \wedge \text{hasRank(pfRank)} = \text{max} \wedge \text{hasItineraryOfferItem(itOff,itOffItem)}$$
$$\Rightarrow$$
$$\text{hasPaymentMode(itOffItem,card)} \vee \neg \exists \text{ a\_card : PaymentCardEnum (hasPaymentMode(itOffItem,a\_card)))}$$

The next formula states that there cannot be a Payment Card with value *min*:

$$\forall \text{ pfRank : PreferenceRank, payCard : PaymentCard, card : PaymentCardEnum}$$
$$(\text{hasPreference(pfRank)=payCard} \wedge \text{hasValueSet(hasOptionValue(payCard))} = \text{card} \Rightarrow \text{hasRank(pfRank)} > \text{min})$$

Category 3: Loyalty Card

- Applied to a single travel episode.
- A travel episode can satisfy **more than one** of the preference values at a time (e.g., a travel episode may allow to use only Millemiglia, only FlyingBlue, or both).
- If a travel episode satisfies more than one preference value at a time, only the value with the greatest score is considered to compute the overall score of the travel episode.
- Scores must be in the (*min*, *max*) interval, thus both *min* and *max* scores are not allowed (i.e. Loyalty Card is only used for ordering).

The following formula states that there cannot be a Loyalty Card with value *max* or *min*:

$\forall$ pfRank : PreferenceRank, ltyCrd : LoyaltyCard, card : LoyaltyCardEnum

(hasPreference(pfRank)= ltyCrd $\wedge$ hasValueSet(hasOptionValue(ltyCrd)) = card

$\Rightarrow$

hasRank(pfRank) < max $\wedge$ hasRank(pfRank) > min)

Category 4

- Applied to a single travel episode.
- This category contains preferences representing **mandatory** requirements.
- Only the *max* score is allowed.
- More than one max score is allowed on the same preference, e.g.:

  score(User=Jane, Preference=PRM-Type, Value=Pregnant) = *max*

  score(User=Jane, Preference=PRM-Type, Value=Deaf) = *max*

- All the retrieved travel episodes must satisfy **every** preference with the *max* score.

The preferences belonging to Category 4 are:

- PRM-Type
- PRM-Parameter

The meaning of preferences “Maximum number of steps”, “Escalator”, “Elevator” is, respectively, “no steps/escalator/elevator allowed”. Thus, assigning the *max* score to the “Maximum number of preference” value means that the user does not want steps on her path (similarly for the other values).

Note that most information about a single travel episode is always known (means of transportation, carrier, seat, class, etc.). However, for the PRM preferences, it might happen that we **do not know** whether a travel episode supports it or not. Travel episodes with unknown information are given a default **dislike** score by the system. In fact, since we have no information, they should have a lower rank than the ones that are surely feasible. However, they should not be discarded because they might turn out to be feasible (e.g. after a phone call). A special alert should be used to inform the user of this situation.

5.2 SCORING THE RESULTS OF QUERIES

When offers are retrieved, each of them is assigned a rank obtained by combining the ranks of the single preferences. In particular:

- Offers containing travel episodes with a *min* score are discarded.

- For preferences in Category 2: Any offer that contains a travel episode that allows online payment but does not allow the use of any of the cards with the max score is discarded.
- The remaining offers must be ordered according to the following criteria (where the aggregation functions could vary depending on the TC):
  - If an offer is composed of multiple travel episodes, the score of a preference defined on travel episodes (e.g., seat) for that offer is computed by applying an aggregation function (e.g., weighted average) to the scores of the travel episodes. If a travel episode does not support a preference (for example, it is not possible to choose the seat on metro), the score of the preference for that travel episode is the indifference score ((*min* + *max*)/2).
  - If multiple preferences on an offer are defined (e.g., preferred carrier, class and seat), the offer is assigned a score through an aggregation function (e.g., arithmetic average).

5.3 TRIP TRACKER PREFERENCES

After buying a trip, users can select what messages about it they would like to receive. For example: Cancellation messages; Rerouting messages; Platform changes; Air conditioning / heating out of order and so on. All (and only) the chosen types of messages will be sent to users.

This type of preferences differs from the categories shown so far as it is neither used to filter offers nor to order them, but, rather, to specify the preferred messages on the already bought offers. TT preferences are semantically similar to the preferences in Category 4. To uniform this category with the others we can say that every type of message users choose to receive can be interpreted as a preference having *max* score, as in the following examples:

- score(User=Jane, Preference=TTPreference, Value=Cancellation) = *max*
- score(User=Jane, Preference=TTPreference, Value=Rerouting) = *max*

6 REFERENCES

[1] IT²RAIL consortium. Deliverable D5.2: Travel Companion Specifications.

[2] IT²RAIL consortium. Deliverable D5.1: Travel Companion Ontology.

End-of-document