Formal description of non-functional service properties

Justin O'Sullivan, David Edmond and Arthur H. M. ter Hofstede

Business Process Management Group
Centre for Information Technology Innovation
Queensland University of Technology
GPO Box 2434
Brisbane QLD 4001 (Australia)
justin.osullivan@gbst.com

February 9, 2005

Abstract

Service discovery is the matching of the needs of a service requestor with the offerings of service providers. It is currently functionally focused (i.e. concentrating on what the service does), and occurs in what can only be considered a volatile and heterogeneous environment. Bring to that equation the area of web services, with its various specifications, and service requestors are no better equipped to efficiently discover both traditional and web services. We offer an approach to describing the non-functional properties of services (including payment, price, availability, obligations, rights, security, trust, quality, discounts, and penalties) that does not differentiate between traditional and web services. It attempts to describe the non-functional properties in a domain independent manner that is motivated by the desire to provide sophisticated service discovery, service selection, automated service negotiation and dynamic service substitution.

Contents

List of Figures

List of Tables

1 Introduction

Service discovery is the matching of the needs of a service requestor with the offerings of service providers. It is currently functionally focused (i.e. concentrating on what the service does), and occurs in what can only be considered a volatile and heterogeneous environment. Bring to that equation the area of web services, with its various specifications, and service requestors are no better equipped to efficiently discover both traditional and web services. Subsequently, the interactions that occur with service providers are drawn-out, challenging affairs that leave most service requestors wondering where the word "service" applies.

Issues that arise for service requestors include the inability to compare services objectively, insufficient details about either or both the functional and the non-functional properties of services (e.g. price, payment, temporal and locative availability), no common language for the description of services (i.e. different service providers use or misuse service related terms), and a lack of mechanisms to assist with effective filtering of relevant services. So how do you find a service that meets your needs?

1.1 Motivation

We believe that services need to be well-described. The descriptions must be semantically rich and they must have structure. We want to exploit that structure to enhance the discovery of services. To be clear, we view the description of a service as consisting of two halves. The first provides a description of the functionality of the service (i.e. what it does), the latter provides a description of the non-functional properties of the service (i.e. anything that exhibits a constraint over the functionality). An example of a non-functional property of a service is its temporal availability. One dimension of temporal availability for services can be considered the constraining of when a service can be accessed by a service requestor.

All these factors, but in particular the lack of description of non-functional service properties, hinder the support for sophisticated service discovery, service selection, automated service negotiation and dynamic service substitution. Our intent is to reduce the "fulfillment gap" that is generated by incomplete information and an internal understanding gap caused by a communication disconnect [37]. The primary motivation behind our research is to utilise this semantic richness of service descriptions to enhance service discovery and substitution. We are interested in the description of these constraints that are exhibited over services. We consider that these constraints may be imposed by the service provider or by the environment within which the service operates.

Our aim is to develop an abstract syntax for a language. As such it exhibits no commitment to items such as keywords or ordering. We do not envisage that our abstract syntax will be capable of describing all services. We prefer not to distinguish between electronic and traditional services. This is because we may request them electronically and receive them traditionally, or vice-versa. We feel that the description of services must cater for both traditional and electronic services.

1.2 Outline

Section 2 provides a set of criteria that we use for including non-functional properties in our language. Section 3 provides a set of models relating to time and locations with respect to availability, payment, price, discounts, rights, obligations, security, quality, trust and penalties. The models express the non-functional properties both directly and indirectly. This is because these non-functional properties will vary depending on the expectations of the service requestor (e.g. trust). Section 4 utilises the models to present a detailed example of a service described in our language. Section 5 discusses issues surrounding the models and their use. Finally, section 6 discusses related work before presenting our conclusions in section 7.

Examples within the paper have been taken from numerous domains as a means of validating the domain independence of the models that we present. Table 1 expresses just some of the domains that our examples are sourced from.

1.2.1 Example service

Traditional services are rich sources of non-functional properties. We present a detailed example here as means of highlighting the type of information that we are trying to capture with respect to services. The example we depict here is that of the Australian ballet company that is to present a production of "Swan Lake". The terms used within the advertisement have been left unaltered.

- ◇ Venue: Lyric Theatre, Queensland Performing Arts Centre, Brisbane.
- ◇ 24 February - 2 March. 7:30pm 24, 25, 26 February and 1, 2 March. 1:30pm 26 February. 6:30pm 28 February.

Domain	Service
Media	Newspaper home delivery
Tourism	Accommodation
Leisure	Entertainment
Housing	House Building
Franchises	Dog Minding
Education	Property Investment Seminar
Transportation	Bus
Maintenance	Carpet Cleaning
Telecommunication	Mobile
Finance	Investment Syndicate
Religion	Church Service
Health	Single Parents Support Service

Table 1: Domains of service examples.

- ◇ Bookings: In person (All Qtix Outlets or the Queensland Performing Arts Centre Box Office. By phone (136246 All major credit cards welcome). Online (www.qtix.com.au).
- ◇ Prices:
  - Adults - Premium ($85), A Reserve ($75), B Reserve ($65), C Reserve ($50), D Reserve ($25).
  - Pensioners - Premium ($75), A Reserve ($65), B Reserve ($55), C Reserve ($40), D Reserve ($25).
  - Youth (under 26 years) - Premium ($60), A Reserve ($45), B Reserve ($35), C Reserve ($35), D Reserve ($25).
  - Children (under 17 years) - Premium ($50), A Reserve ($35), B Reserve ($30), C Reserve ($30), D Reserve ($25).
- ◇ Further details available fom australianballet.com.au.

We consider this style of descripton to be typical of traditional service descriptions (i.e. advertisements). In section 4 we present a further discussion on how this example could be supported within our language.

2 Language criteria

Any language should be assessable according to a series of criteria. We base the criteria for our language on the requirements of information modelling techniques [16] and the roles of knowledge representation outlined in [11]. These criteria can be summarised as formal (providing grounding for reasoning and validation); expressive; executable (ability to transform from abstract to concrete form); comprehensible (capable of being understood by humans); conceptual (separation of concepts in language and its implementation); and suitable (close relationship exists between domain concepts and elements of the language).

During our research we've regularly applied additional criteria to determine if a non-functional property should be included within the language. We consider these criteria a refinement of the "suitability" criterion mentioned above. Our additional criteria include:

- ◇ Does the example being used to test the model present itself in numerous services across multiple domains. We cannot expect to represent every service. An approach that tries to support such a range of non-functional properties will inevitably become cumbersome. How much specification effort is required? Do the benefits outweigh the cost/effort involved? An example of this test is the use of entities within our model that relate to conditions and procedures. Whilst capturing useful information the effort involved in capturing their domain independent properties is significant.
- ◇ We ask ourselves "Have we stepped into a specific domain by including a certain non-functional property within our model(s)?". If so, the non-functional property is excluded from use within our models. For example, the writing of personal bank cheques in some countries results in the originator of the cheque being charged a government levied charge or tax. We do not model the specific charge name but are still able to capture the additional charges that may be imposed on the use of the service. We attempt to remain domain independent in our approach.
- ◇ Does the abstraction of a concept from the Universe of Discourse result in the loss of semantically useful information (i.e. Does it limit the conceptual queries that we can apply to a model)? If the querying of the

model is limited by providing an abstraction for a concept then we do not pursue that abstraction. This criterion typically applies when determining subtype hierarchies. Common properties are attached to the supertype with only the relevant properties being attached to the subtype. If by including the property as a generically named role in the supertype then we have made the querying of the model more difficult. This sometimes comes at the cost of not having a pure subtype hierarchy.

◇ Does the example being tested require knowledge that is not available to the service provider at the time of description? The process of publishing a description to a service catalogue must be done without context relating to the service requestor. For example, we cannot state the price of a service with specific discounts included as discounts require knowledge about the service requestor (e.g. age). We therefore state price and discounts separately, expecting that the service catalogue will take a more active role in the discovery process by using information that it may know about the service requestor and applying it to the discovery of information within the catalogue.

◇ A primary area of interest for future case studies are the areas of service substitution and composition. Does the model still hold (roughly) under the application of composition and substitution? Discovery is viewed as fundamental to the process of querying required for service substitution and composition.

3 Non-functional property models

There are numerous concepts that underpin an abstract language such as ours. It has been necessary to capture common concepts such as time (i.e. temporal), place (i.e. locative), price, payment, discounts, obligations, penalties, rights, security, trust and quality that we regularly encounter independently of services. We have then tried to put these into the context of services. The sub-sections that follow present a series of models that we utilise to support our description of the non-functional properties of services.

We have relied upon the use of Object-Role Modelling (ORM) as a mechanism for portraying the taxonomy that we present herein. ORM is a fact-oriented modelling technique that makes no use of attributes. All facts are represented in the form of entities playing roles. A detailed discussion of ORM is presented in [15]. Looking at Figure 1 we can see ORM entities are represented as named ellipses (e.g. Service, Provider). Attached to the entities are roles that provide a description of the part that an entity plays in a relationship. These relationships, or associations, are represented using one or more role boxes. These are referred to as unary, binary, ternary relationships depending on the number of role boxes. Roles are attached to Entity or Value objects using solid lines. Where they appear with a black dot at the point of connection between the line and the entity/value object, then these roles are considered mandatory. Patterned ellipses refer to entities that are defined elsewhere (in another model). We have attempted to provide a partial population of the models as an aid to their understanding. We also make use of ConQuer, a conceptual query language, within this paper as a means of providing examples of conceptual queries that could be applied to the ORM schemas. A detailed discussion of ConQuer is presented in [6].

3.1 Service provider

We use this section to present our first model relating to the service provider [see Figure 1]. We stipulate that service providers have functional offerings(i.e. services) for service requestors. The function offered is referred to as a “Capability” [22] and each service provides one capability to requestors. We consider the area of capabilities to be the boundary of our work with respect to the functional perspective of services. We are not attempting to provide a functional description of a service. Our one to one relationship between service and capability is largely motivated by a reduced specification effort and an attempt to compel service providers to describe their services in a fine-grained manner. Introduction of a one to many relationship enforces the need to offer specialisation mechanisms that cater for inheritance of non-functional properties such as service availability, price and payment. If inheritance were to be available, non-functional properties would need to be able to be specified at both the service and the capability level. This aggregation of capabilities into services must then be sufficiently expressive to cater for overriding and exclusion of non-functional properties between the service and capability.

We envisage that a service provider will probably be an organisation (in the generic sense of the word), but may alternatively be multiple organisations. Providers are internally identified by a unique identification scheme. Dun and Bradstreet’s D-U-N-S number is one possibility for this internal identification scheme [12]. Providers utilise a name for general identification. These provider names are generally granted to a provider by a regulatory authority. It is included in our model since organisations are more commonly referred to using a provider/company name (e.g. Microsoft Corporation, Deutsche Bank). We consider each service to operate within one or more service industries that are identified according to a United Nations Standard Products and Services Code (UN/SPSC) [21].

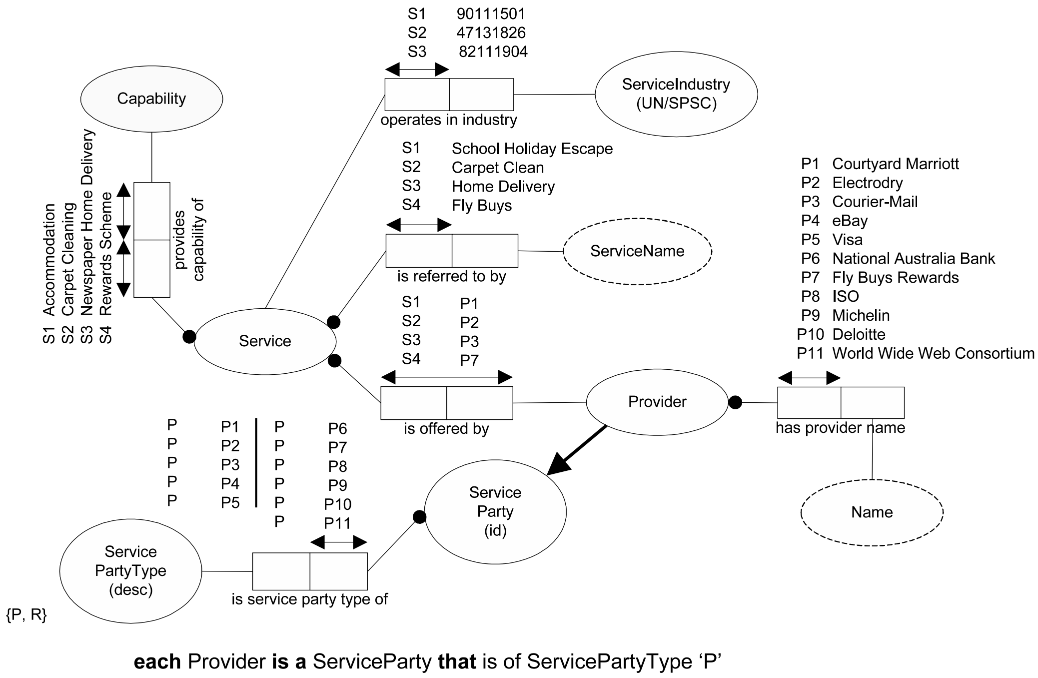


Figure 1: Service provider.

Services are referred to by their service name. Names for services are normally intended to be unique across service providers. However, they are sometimes duplicated across services in different domains. A service name is normally determined by the service provider. The combination of a service name and a provider identifier is normally sufficient to identify a service. An exception to this is when multiple service providers are involved in the provision of a single service.

Figure 1 presents a single formal subtype definition (i.e. each Provider is a ServiceParty that is of ServicePartyType 'P') that is an ORM mechanism for determining membership of subtypes, using the service party type role on the ServiceParty supertype. The additional enumeration constraint of 'R' is used to define a Requestor service party. As there are no specific roles to be attached to the Requestor subtype it is not depicted as an entity in the diagram. We include it here for use in a later discussion with respect to rights.

3.2 Temporal model

The first non-functional property model that we have chosen to present is a temporal model. We use this model as the foundation for capturing the temporal availability of a service (i.e. the "when" aspects of a service) in section 3.4. This section attempts to define the types of temporal concepts that will be required for service description. Temporal concepts are regularly used within service descriptions to represent such things as when a service can be requested, provided, or queried for further information. Examples of temporal descriptions include:

- Newspaper delivery service - A newspaper provider is offering a home delivery service for two newspapers. This home delivery service can be requested from 7am to 8:30pm Monday through Friday, and 9am to 2pm Saturday.
- Accommodation service - A hotel on the Gold Coast advertises a special or discounted rate. The rate is applicable to the accommodation being provided between 17th September and 10th October 2004 (inclusive).
- Entertainment service - A seafood festival is provided from 5:30pm to 12 midnight on Friday 27th August 2004. It is part of a larger festival which occurs in Brisbane from 27th August to the 4th September 2004.
- Home building service - A home building company offers a display village in five locations around a city. The first location is open from Thursday through Sunday 10am to 5pm, and Monday 2pm to 5pm. The second location is open from Tuesday through Sunday from 10am to 5pm, and Monday 2pm to 5pm.

The third location is open from Thursday through Sunday 10am to 5pm, and Monday 12pm - 5pm. The remaining two locations have the same opening times as the second location.

We collectively refer to all types of temporal concepts as “temporal entities” [see Figure 2]. We provide for the description of four primary types of temporal entities - dates, time (both anchored and recurring representations), intervals, and durations. We provide a discussion of these types in the sub-sections that follow. Firstly we present dates and time as they act as the basis for the description of instants, which in turn are used to describe intervals.

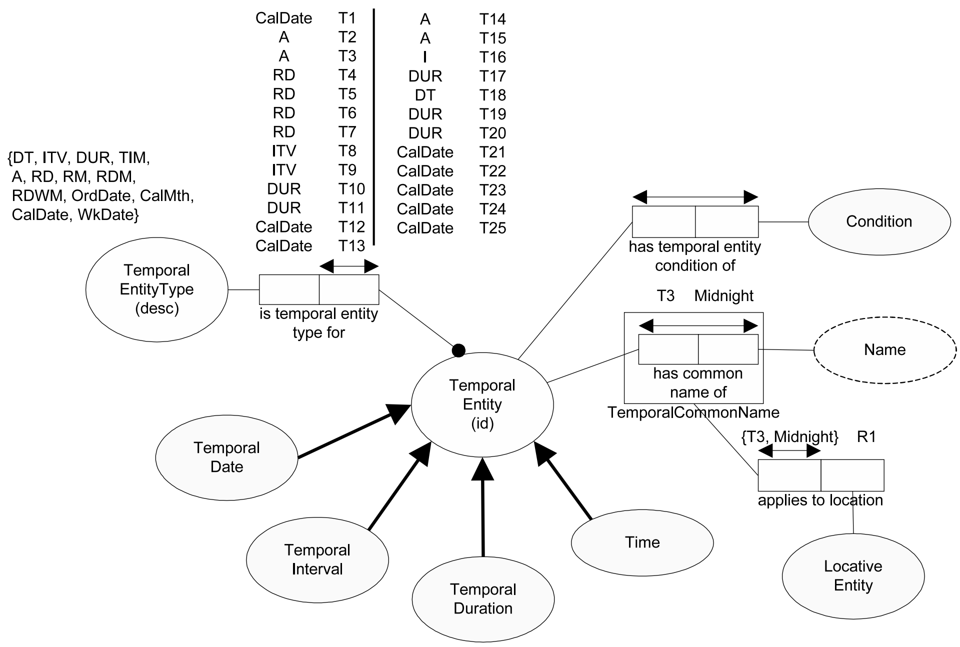


Figure 2: Temporal entities.

We provide for the expression of a common name for a temporal entity. This allows us to express a temporal concept such as an interval (e.g. a particular week of the year) and apply a common name to assist with service discovery. For example, in Australia the first Sunday of September is Father’s Day. Our temporal model provides for the expression of the temporal interval, the first Sunday in September (more details are provided later). To this temporal entity that we describe we can then attach the common name “Father’s Day”. Additionally, we offer the ability to put this temporal common name into a locative context. This locative context is likely to be a region such as a country or state.

3.2.1 Temporal dates

We classify dates into four subtypes - calendar dates, ordinal dates, week dates and calendar months [see Figure 3]. The first three of these are outlined in [14]. Our discussion of dates assumes the use of the Gregorian calendar. The four date subtypes can be defined as:

- *Calendar dates* are described using a year number, a month number between 1 and 12 (representing the months of January through December) and a day of month number between 1 and 31. For example, a year number of 2004, a month number of 8 and a day of month number of 27 represents the 27th August 2004.

- *Ordinal dates* are a combination of a day of the year number between 1 and 366 (catering for leap years) and a year number. For example, a day of the year number of 240 combined with a year number of 2004 represents the 27th August 2004.

- *Week dates* are defined using a day of week number (where 1 - 7 identifies the days Sunday through Saturday), a week of year number (indicated by a number between 1 and 52), and a year number. For example, a day of week number of 6, a week of year number of 35, and a year number of 2004 represents the 27th August 2004.

- *Calendar month dates* are a subset of the properties of calendar dates, and are defined using a month of year number and a year number. For example, a month of year number of 8 and a year of 2004 represents August 2004.

Figure 3 includes an ORM external uniqueness constraint that stipulates uniqueness across two or more fact types. The external unique constraint is depicted as dotted lines joined to a circled "U". This figure also includes a derived role. The role "is a leap year" is attached to the Year entity. The "*" denotes that it is derived. In this case, the derivation is based on a well-known algorithm to determine if the year is a leap year or not.

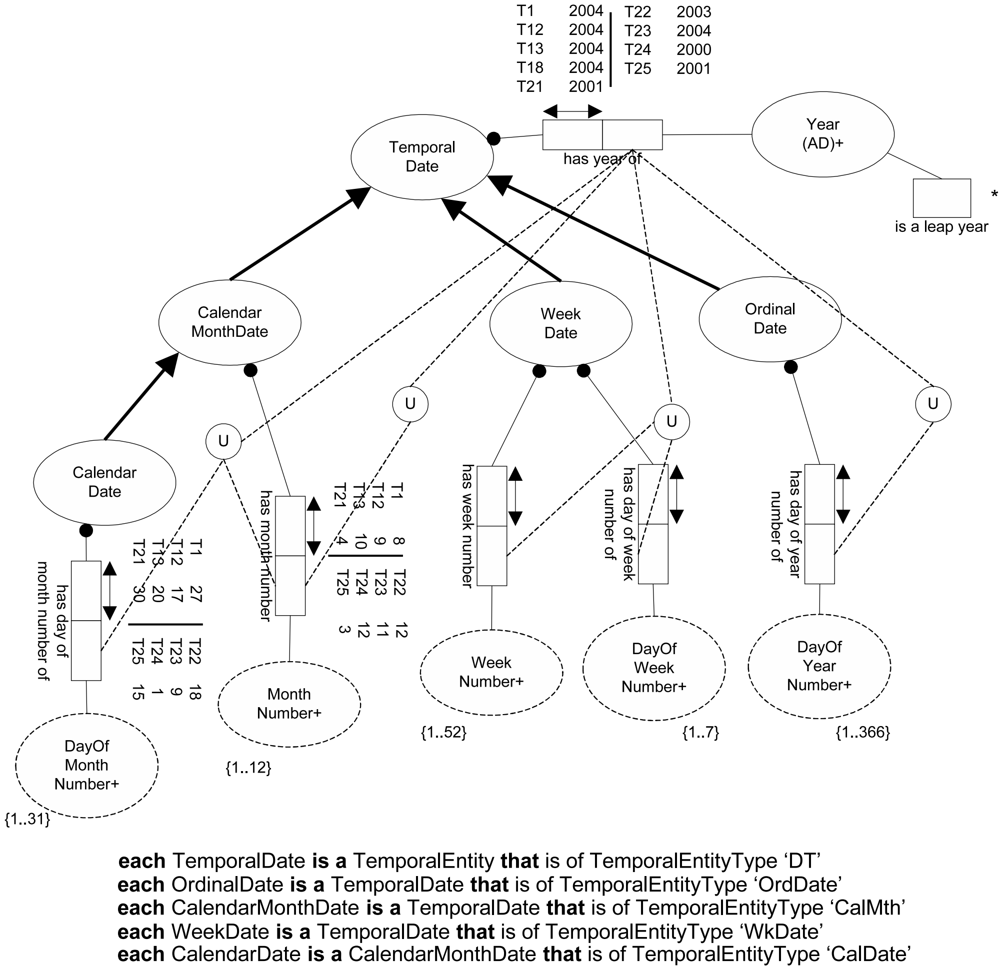


Figure 3: Temporal dates.

The following is an example ConQuer query over week dates that returns all the TemporalEntity instances where the year is 2004, the week number is 39, and the day of the week number is 6. Assuming that the first week of the year 2004 starts on Sunday 4th January, then the following query returns all instances equal to the 1st October 2004.

```
TemporalDate
  ⊢ has year 2004
  ⊢ has WeekDate
      ⊢ has week number 39
      ⊢ has day of week number 6
```

The following is an example query over calendar dates that returns all the TemporalEntity instances where the year is 2004, the month number is 8 and the day of the month number is 27. This returns all instances equal to the 27th August 2004.

```
TemporalDate
  ⊢ has year 2004
  ⊢ has CalendarMonthDate
```

⊢ has month number 8
⊢ has CalendarDate
⊢ has day of month number 27

3.2.2 Time

We represent the concept of time for use within the description of points in time. By time we refer to the time of day that is displayed on a clock. When combined with other properties (e.g. a calendar month date, a day of the week) we consider points in time to be either anchored (occurring only once), or recurring (occurring more than once, normally with some regularity) [see Figure 4]. Time is particularly useful for the description of intervals (either the start or the end of an interval in combination with a duration, or jointly), and as a deadline (e.g. when payment of a bill or invoice is due).

When used to describe a temporal interval, points in time act as boundary positions (i.e. the start or end position) of an interval (see section 3.2.3). Accordingly, we believe that we achieve a similar notion of anchored versus recurring temporal intervals through the use of recurring points in time. We do not attempt to describe time with a granularity smaller than seconds (e.g. milliseconds). For other parts of the model that describe temporal durations we provide for the ability to define granularities of less than a second. We therefore say that our time representation cannot be further divided into a smaller unit of time.

We consider all times to have at least the following properties: hours (a unit of 60 minutes), minutes (a unit of 60 seconds), seconds (the smallest unit of time that we choose to represent), and a timezone (expressed as a positive or negative offset from -12 to + 12). This offset is expressed according to Coordinated Universal Time (UTC), and includes hours and minutes (e.g. Australian Central Daylight Time is UTC +10:30). A full list of UTC timezones is available from [28]. In addition to the offset we capture a region that the timezone relates to. This is via a type of locative entity called "Region" that will be discussed in more depth in section 3.3.3.

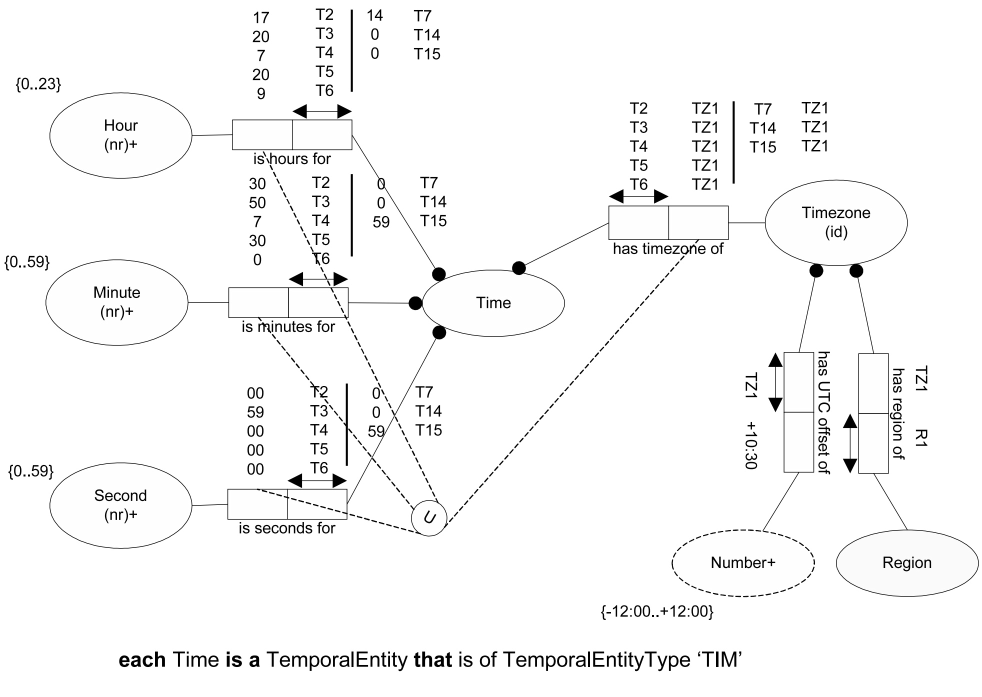


Figure 4: Time.

As mentioned previously, we offer the ability to store a common name for all temporal entities. The usefulness of this in a time context is visible for describing a UTC time of 00:00:00+10:30 using the common name "midnight", whilst 12:00:00+10:30 could be given the common name "midday".

Points in time We believe that time has five specific subtypes, the most important of which is anchored points in time. This will be discussed shortly. Other subtypes of time include recurring daily time, recurring day of month time, and recurring day of week in month time. Examples of these recurring time subtypes include:

◇ Recurring Daily Time in a Week - Day of week number and a time. For example, a day of week number of 2 and a time of 14:25:00+00:00 represents Monday at 2:25pm.

◇ Recurring Daily Time in a Month - Month number and a time. For example, a month number of 8 and a time of 06:00:00+00:00 represents every day in the month of August with a time of 6:00am.

◇ Recurring Day of Month Time - Month number, day of month number and a time. For example, a month number of 11, a day of month number of 27 and a time of 18:30:00+00:00 represents November 27th at 6:30pm.

◇ Recurring Day of Week in Month Time - Occurrence number, day of week number, month number, and a time. For example an occurrence number of 1, a day of week number of 2, a month number of 11 and a time of 09:45:00+10:30 represents the 1st Monday in November at 9:45am (with a UTC offset of 10 hours and 30 minutes).

Anchored points in time Anchored points in time are fixed by inclusion of a date with the time [see Figure 5]. As stated previously, anchored points in time occur only once. This subtype allows us to capture requirements such as due dates (e.g. for payment) and to build temporal intervals using a start and end anchored point in time. All Time subtypes inherit all the properties of the Time temporal entity (i.e. hour, minute, second and timezone). The attachment of a date to a time produces a temporal instant. Importantly, temporal instants include a date where the type of date used has a "day" level of granularity, not a particular month as is the case with calendar month date types. See section 3.2.1 for a discussion of temporal dates.

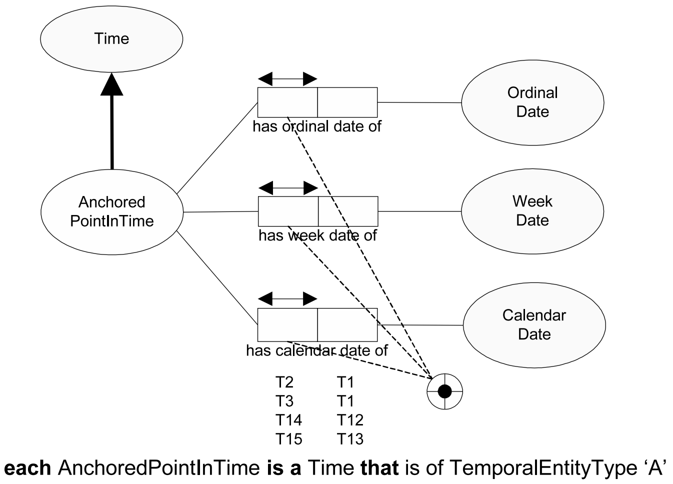


Figure 5: Anchored temporal instants.

Figure 5 includes an ORM exclusive-or constraint that stipulates only one role is played. The exclusive-or constraint is depicted as dotted lines joined to a circled "X" with a mandatory symbol (solid black dot) centred over the "X".

The following is an example ConQuer query over an anchored point in time. The query returns all the times where the time is 8:30am on the 6th February 2006 in any timezone that is specified using a CalendarDate.

```
AnchoredPointInTime
  ⊢ has hours 8
  ⊢ has minutes 30
  ⊢ has seconds >= 0
  ⊢ has seconds <= 59
  ⊢ has CalendarDate
      ⊢ has year 2006
      ⊢ has month number 2
      ⊢ has day of month number 6
```

Recurring times As previously mentioned in this section we believe that the ability to describe the following four types of recurring times is useful in the context of service description: recurring daily time in a week, recurring daily time in a month, recurring day of month time, and a recurring day of week in month time. We refer to all these subtypes as not being anchored in time, and can therefore say that they apply at multiple anchored points in time over an anchored temporal interval [see Figure 6]. All subtypes inherit the properties of the time entity (i.e. hour, minute, seconds and timezone).

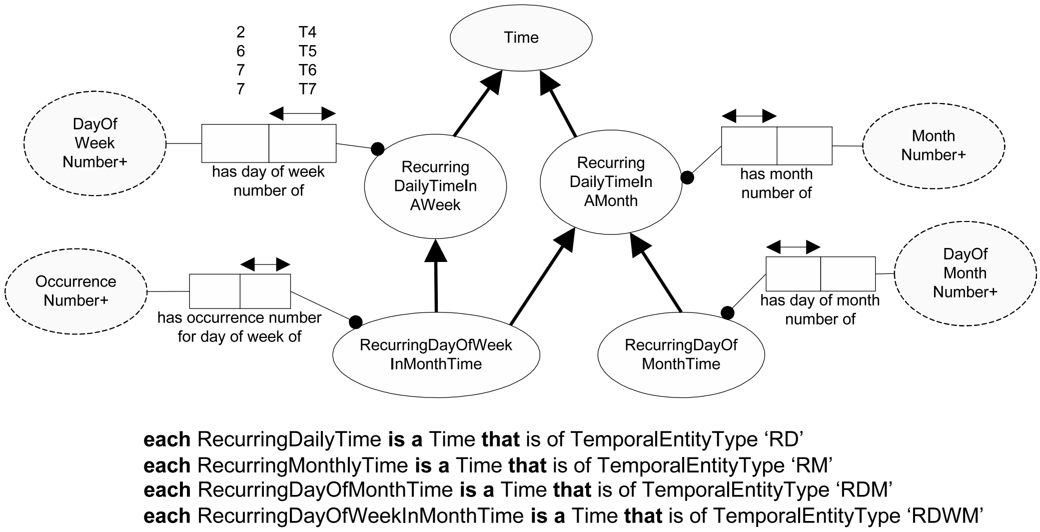


Figure 6: Recurring time.

As its name suggests, recurring daily time in a week is useful for describing a time that occurs on a daily basis. For example, Monday 9am might be used to describe the time that provision of a service regularly starts. In the newspaper home delivery example provided above, the service was able to be requested from 7am to 8:30pm Monday through Friday, and 9am to 2pm Saturday. We may also choose to think of this as being 7am - 8:30pm Monday, 7am - 8:30pm Tuesday and so forth until Friday, and then 9am to 2pm Saturday. Each case is an interval that can be demarcated by a start and an end time that is of a recurring daily time in a week type.

Recurring daily times in a month specify a month number. This means that the recurring nature of this time subtype is daily within the month specified.

Recurring day of month times specify a day of the month, and a month. For example, a local church may offer a Christmas mass where the service is provided each year on the 25th December at 6:00am. This means that the recurring nature of this subtype is annually.

Recurring day of week in month times specify the occurrence of a day of week, within a particular month. For example, a single parents support group may meet on the 1st Monday of each month at 7:00pm. The corresponding population of this subtype requires that each of the twelve months of the year be specified (i.e. 1st Monday of January at 7:00pm, 1st Monday of February at 7:00pm through to the 1st Monday of December at 7:00pm).

3.2.3 Temporal intervals

We refer to intervals as being bounded by times that are either anchored or recurring. Providers historically have used recurring intervals to describe the availability of their services. The length of time between the start time and the end time of an interval we refer to as the temporal duration. In the case of a recurring interval the duration is the length of time between the start and end of a single instance of the recurring interval. We express temporal intervals using one of three different approaches [see Figure 7]. Each of the following approaches are outlined with an example of anchored points in time:

- ⋄ From a Time to another Time: This type of interval is defined with a specific start time and end time. For example, a start time of 9:00:00+10 and an end time of 23:00:00+10:00 defines the period between 9am Australia Eastern Standard Time and 11pm (in the same timezone). The second time that is described should occur temporally after the first time. An example of this type of interval is the entertainment service outlined in section 3.2 that began at 5:30pm on the 27th August 2004 and finished at midnight on the same day.

- ⋄ From a Time for a Duration: This type of interval is described by declaring a start time and specifying a duration. For example, a start time of 8:30:00+10:00 on calendar date 20/06/2004 with a duration of 3 days. The end time can be derived by adding the duration to the start time. In this case the end time would be 8:29:59+10 on the 23/06/2004.

◇ For a Duration to a Time: The reverse of the previous approach can be used to describe a duration with a specific end time. Using this approach the start time can be derived from taking the duration away from the end time.

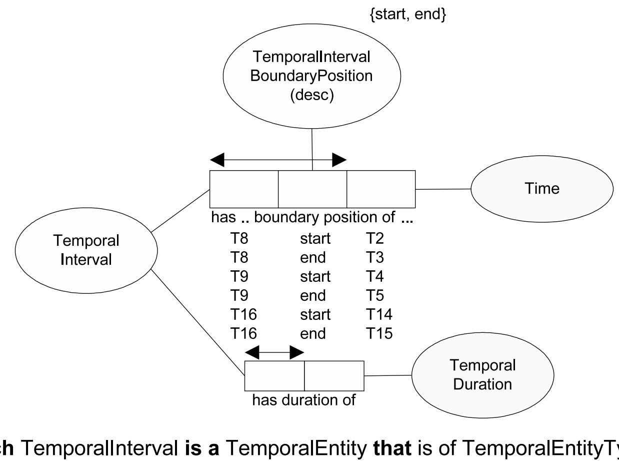


Figure 7: Temporal intervals.

It is easy to understand that dates are a form of temporal interval. A particular date (e.g. 10/03/2005 or 10th March 2005) can viewed as: time to another time (0:00:00+10:00 10/03/2005 to 23:59:59+10:00 10/03/2005), time with a duration (0:00:00+10:00 10/03/2005 with duration of 1 day), or duration to a time (1 day to 23:59:59+10:00 10/03/2005). We provide a distinct treatment of dates within our temporal model. See section 3.2.1 for further details.

The following is an example ConQuer query over a temporal interval using anchored points in time that have been defined with a CalendarDate. The query returns all the instances where the start time is 8:30:00+10:00 on the 6th February 2006, and the end time is 8:30:59+10:00 on the 9th February 2006.

TemporalInterval
  ⊢ has 'start' boundary position of AnchoredPointInTime
    ⊢ has hours 8
    ⊢ has minutes 30
    ⊢ has seconds 0
    ⊢ has UTC offset +10:00
    ⊢ has CalendarDate
      ⊢ has year 2006
      ⊢ has month number 2
      ⊢ has day of month number 6
  ⊢ has 'end' boundary position of AnchoredPointInTime
    ⊢ has hours 8
    ⊢ has minutes 30
    ⊢ has seconds 59
    ⊢ has UTC offset +10:00
    ⊢ has CalendarDate
      ⊢ has year 2006
      ⊢ has month number 2
      ⊢ has day of month number 9

Some providers prefer to use temporal interval descriptions such as 9am till late. This could be facilitated within our model by allowing the provider to specify an end instant for the temporal interval, and assigning it a temporal common name of "late".

Recurring temporal intervals Expressing recurring temporal intervals is useful for service providers who wish to regularly advertise the availability of a service without the need to update a service description. For example, the newspaper delivery service was available from 7am to 8:30pm Monday through Friday, and 9am to

2pm Saturday. We could use the notions of recurring daily time in a week presented previously in conjunction with an interval in the following way to represent this example. Remembering that this example is really the conjunction of 6 intervals, the first interval start time could be represented using a recurring daily time of Monday 7am, with a terminating time of 8:30pm Monday. Alternatively, the same start time could be represented with a duration of 13.5 hours.

We envisage the need to express the following types of temporal intervals:

- A month (e.g. December): This could be represented as a recurring day of month time of (1st December at 0:00:00+00:00 with a duration of 31 days).
- An occurrence of a day of week within a month (e.g. the 3rd Sunday in July): This could be represented using the RecurringDayofWeekInMonthTime that has the values - the month number is 7 (for July), the day of week number is 1 (for Sunday), the occurrence number for the day of week is 3 and the time is 0:00:00+00:00. With a duration of 24 hours this interval is capable of being represented.
- A day of a month (e.g. 25th December): This could be represented with a day of month number of 25, and a month number of 12, and a time of 00:00:00+00:00. Like these other examples the interval can be expressed with this start time and the duration (in this case 24 hours), the start time and an end time (23:59:59+00:00 on the 25th December) or a duration and the end time.

The following is an example ConQuer query over a recurring temporal interval such as day of a month. This query returns all TemporalIntervals that represent the recurring temporal interval of the 25th December using a start time and a duration. Alternatively, this could be queried using the common name attached to the recurring temporal interval.

```
TemporalInterval
  ⊢ has 'start' boundary position of RecurringDayOfMonthTime
      ⊢ has hours 0
      ⊢ has minutes 00
      ⊢ has seconds >= 0
      ⊢ has month number 12
      ⊢ has day of month number 25
  ⊢ has duration of TemporalDuration
      ⊢ has cardinality 24
      ⊢ has temporal granularity of StandardTemporalGranularity
          ⊢ has standard granularity name 'Hour'
```

Temporal interval operations We provide the ability to recursively describe exceptions to intervals (i.e. exclusion of a sub-interval) and restrictions over an interval (i.e. refinement to a particular sub-interval) [see Figure 8]. This allows us to specify a temporal interval (e.g. June - November 2004) but to restrict to just a subset of that (e.g. Tuesday's) within that interval.

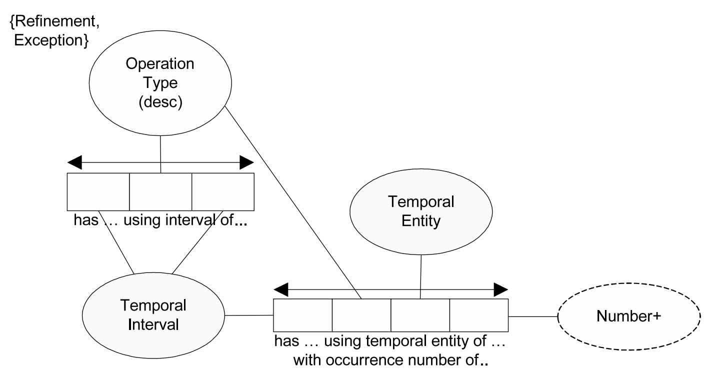


Figure 8: Temporal interval operations.

We offer an alternative to specifying the refinements and exclusions within a temporal interval. This is depicted in Figure 8 as the quaternary relationship whereby a TemporalInterval entity can have an OperationType (i.e. refinement or exclusion) specified as another TemporalEntity with a particular occurrence number within the interval. This allows us to perform refinement and exclusion such as, specifying the occurrence of a particular day of the week within a month (e.g. 2nd Sunday in August 2005).

We build upon our previous example query to show the use of temporal interval operations in the context of searching for exclusions to service availability (i.e. a service is not available on the 25th December). This query returns all TemporalIntervals that represent the recurring day of month time temporal interval of the 25th December as an exception.

TemporalInterval
  ⊢ has OperationType of 'Exception' using interval of TemporalInterval
    ⊢ has 'start' boundary position of RecurringDayOfMonthTime
      ⊢ has hours 0
      ⊢ has minutes 00
      ⊢ has seconds >= 0
      ⊢ has month number 12
      ⊢ has day of month number 25
    ⊢ has duration of TemporalDuration
      ⊢ has cardinality 24
      ⊢ has temporal granularity of StandardTemporalGranularity
        ⊢ has standard granularity name 'Hour'

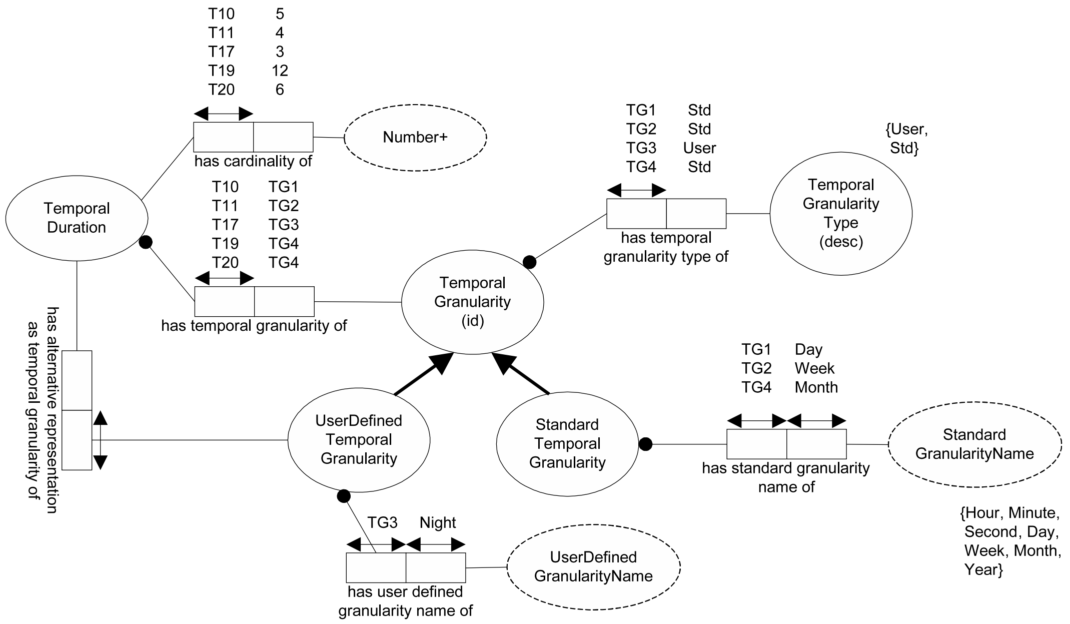
Alternatively, an interval could be represented as finishing prior to the 25th December or beginning after the 25th December. The union of another query would be required to produce all intervals that exclude the 25th December.

The following example query outlines the use of temporal interval operations for refinement. The query presents the refinement of an initial temporal interval (e.g. August 2005), to be the 1st Sunday in August 2005 at 10am.

TemporalInterval
  ⊢ has 'start' boundary position of AnchoredPointInTime
    ⊢ has hours 0
    ⊢ has minutes 00
    ⊢ has seconds >= 0
    ⊢ has month number 8
    ⊢ has day of month number 1
    ⊢ has day of month number 2005
    ⊢ has duration of TemporalDuration
      ⊢ has cardinality 31
      ⊢ has temporal granularity of StandardTemporalGranularity
        ⊢ has standard granularity name 'Day'
⊢ has OperationType of 'Refinement' using TemporalEntity with occurrence number of 1
  ⊢ has RecurringDailyTime
    ⊢ has hours 10
    ⊢ has minutes 00
    ⊢ has seconds >= 0
    ⊢ has day of week number 1

3.2.4 Temporal duration

The final type of temporal entity, temporal durations are used to express lengths of time (e.g. 5 days, or 4 weeks) [see Figure 9]. We describe temporal durations using a cardinality (a number), and a temporal granularity. We divide temporal granularities into one of two different types. Standard temporal granularities are temporal concepts that are readily familiar in most domains. These concepts include hour, minute, second, day, week, month and year. Alternatively, user defined granularities can be captured. These may include notions such as business days. We allow these user defined temporal granularities to be expressed in terms of another temporal duration. Temporal durations may also have a temporal common name attached through the supertype TemporalEntity. This can be useful as it allows us to specify a duration of 1 day and assign it a temporal common name of "Monday". We apply this use of names for TemporalDuration entities when using temporal interval operations (e.g. the refinement or exclusion of the ‘x”th occurrence of Monday within a temporal interval).



each TemporalDuration **is a** TemporalEntity **that** is of TemporalEntityType ‘DUR’
each StandardTemporalGranularity **is a** TemporalGranularity **that** is of TemporalGranularityType ‘User’
each UserDefinedTemporalGranularity **is a** TemporalGranularity **that** is of TemporalGranularityType ‘Std’

Figure 9: Temporal duration.

3.3 Locative model

The next model that we have chosen to present relates to the non-functional property of location. Our notion of the locative aspect of services is wider than just a geographic interpretation. The locative model that we present(i.e. “the where”) acts as a foundation (along with our temporal models) for the availability of a service.

This section attempts to define the types of locative concepts that will be required for service description. Locative concepts are regularly used within service descriptions to represent properties such as where a service can be requested from, where it can be provided to, and where payment can be made. Examples of such descriptions include:

- ◇ Dog minding service - A dog minding service is provided in Brisbane, the Gold Coast and Cairns.
- ◇ Accommodation service - A hotel in Sydney accepts requests on a published phone number, on a toll free phone number and via email using a specific email address.
- ◇ Seminar service - A property investment seminar can be requested using a published phone number and is provided at a specific hotel in Melbourne (address provided).
- ◇ Home building service - A builder provides five locations for display villages. The address of these villages are provided using the street address and a street directory reference. More information is available using a published phone number, or via their web site.

We collectively refer to all types of locative concepts as “locative entities”. We divide locative concepts into the following subtypes: points (stationary and moving), regions, routes, addresses, phone numbers, street directory references, URIs, spectra, internet protocol addresses and ethernet addresses [see Figure 10]. To each instance of a locative entity we allow one or more regionalised, common names to be attached. This is similar to our treatment of TemporalEntity common names.

Locations represent an important non-functional property that is not only capable of describing such things as where a service provider can provide the service to, it also indicates the presence of distance between the requestor and provider. For example, electronic locative entities such as URIs, spectra, IP addresses and Ethernet addresses involve communications as “arms length”. We have previously referred to this as a channel [23]. We now refer to it simply as an interaction.

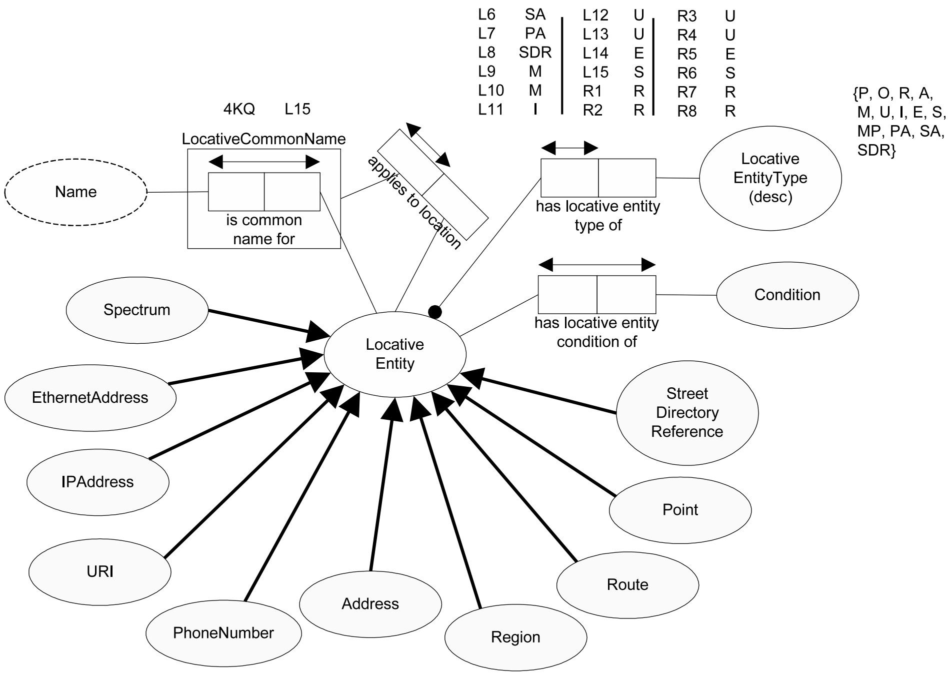


Figure 10: Locative entities.

3.3.1 Point

Points refer to positions on the Earth's surface that can be identified by coordinates of latitude and longitude [see Figure 11]. An altitude may also be recorded for that point. Coordinates are described using degrees, minutes and seconds. Degrees are 1/360th of a circle and are further subdivided into 60 minutes, and then into 60 seconds [13]. In a geographical context, these are sometimes referred to as "waypoints". When we refer to points we are not referring to celestial coordinates.

We further subtype points to include a "MovingPoint" entity. In addition to the properties of stationary points, moving points include a velocity, an anchored point in time and a direction. Moving points are useful when used in the context of routes which are discussed in the section that follows.

3.3.2 Route

We consider a route to be an ordered collection of points. To ensure that the service provider is not burdened with outlining the specific details of the route, we present an abstraction that allows for the specification of a route to be optionally attached. Figures 12 and 13 present our modelling of routes. Routes are commonly used to store the locative properties of moving objects such as buses, trains and planes. We assign a route type and a route name to each route. Together, these act as a high level description of the route. The enumeration constraints presented in Figure 12 for route types are intended to be indicative.

We envisage that the catalogue provider may provide a base specification for major routes, and the service provider may refer to this specification. Alternatively, they may choose to provide their own specification that is either more course-grained or fine-grained.

The ordering of the points within the route is particularly useful for determining the start and end of the route. Routes, like the other subtypes of locative entities, include one or more common names. In a manner similar to our temporal model we allow for the restriction of a route (i.e. refinement to a sub-section of the route) and for exceptions to a route (i.e. exclusions of part of a route). Since a route is capable of capturing the anchored points in time of a series of moving points, the notion is similar to that of a "schedule". Whilst we realise that some routes may involve stopping at the same point multiple times within a single invocation of the service (e.g. a route shaped as a figure of eight), we utilise the same point but provide it with a different ordering value. We apply a frequency constraint to the creation of a route that enforces the need for 2 or more Point instances to ensure the existence of a beginning and an end to the route, otherwise it can only be considered a point. It should be noted that routes can be an ordered collection of MovingPoint entities. This allows for the inclusion of directional and velocity related information in the route description.

The following example query filters for instances of route specifications where the first moving point of the route is referred to by the common name "Brisbane" and where the last point of the route is referred to by the common name "Gold Coast".

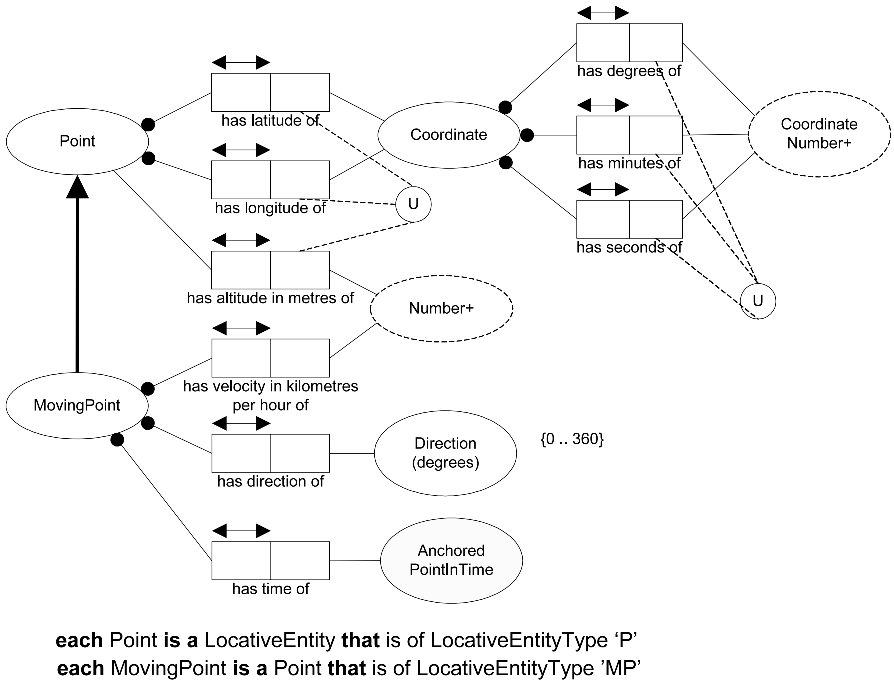


Figure 11: Points (including moving points).

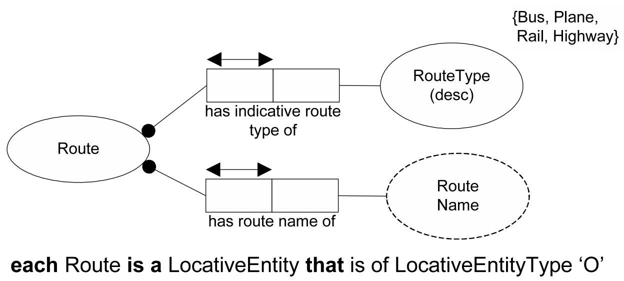


Figure 12: Routes.

RouteSpecification
⊢ has Point
  ⊢ has common name "Brisbane"
⊢ has min(ordering) for RouteSpecification
⊢ has Point
  ⊢ has common name "Gold Coast"
⊢ max(ordering) for RouteSpecification

3.3.3 Region

Our treatment of regions is similar to that of routes. We abstract the specification of the region to reduce the burden of specification on the service provider. Figures 14 and 15 present our modelling of regions. We assign a region type and a region name to each region. Together, these act as a high level description of the region. The enumeration constraints presented in Figure 14 for region types are intended to be indicative.

We consider a region to be a bounded collection of non-moving points which are used to describe an area. Regions are an abstraction that we use for capturing concepts such as countries, republics, states, territories, provinces, counties, cities, and suburbs. Other less common types of regions include franchise areas and amusement parks. We consider RegionSpecification entities to provide the detail about the region. We allow for exceptions to a larger region to be stated, whilst restrictions (or sub-areas) may also be captured. We consider that region specifications are likely to be a part of the service catalogue that will be populated by the catalogue provider. This population of parts of the catalogue ensures that (in this instance) common regions such as

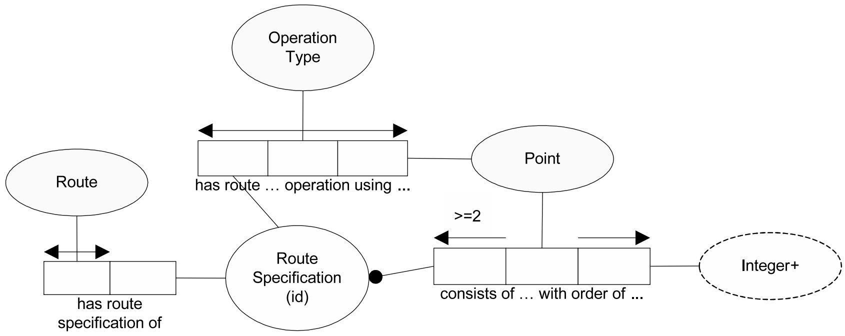


Figure 13: Route specification.

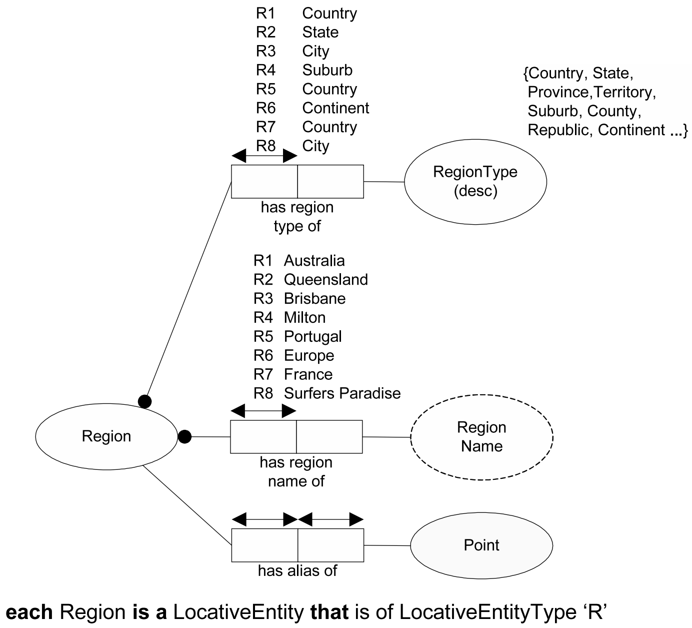


Figure 14: Regions.

countries and states are available for use by service providers.

We assign a frequency constraint to the ternary fact type that is an attempt to enforce a minimum of three points to constitute a bounded region. The authors realise that it is possible to produce a line (with three points along the line) using such a constraint but it is our intent that a bounded region be formed. We are unable to graphically represent in ORM that a region should be a bounded area. To form a line with multiple points we utilise the notion of a route outlined in section 3.3.2.

The following example query filters for instances of regions where the region type is "Country" and the region name is "United States of America".

Region
⊢ has region type of "Country"
⊢ has region name of "United States of America"

3.3.4 Addresses

We consider that addresses are normally of two types: a street address or a postbox address [see Figures 18 and 20]. In general, all addresses include a country, a state or province, a city, suburb, and a postcode or zipcode [see Figure 16].

To each type of address we offer the ability to capture information related to the party at the address to which a request is being directed [see Figure 17]. This addressee related information such as addressee name, professional title, functional title, department name and/or organisation name are presented here, and not in the service provider model (see section 3.1) for one particular reason. When we describe services, the

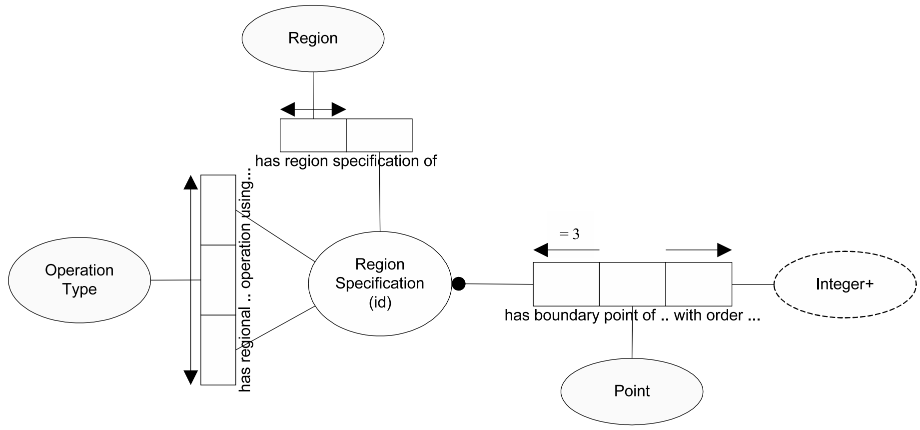


Figure 15: Region specifications.

interactions that occur between a service provider and a service requestor can only happen at a location, or via a communication mechanism to a location. This addressee related information is specific to the address subtype of LocativeEntity. This is accurate whether the address is used to reflect a postal address or to indicate the physical presence of the service provider.

Street addresses may include the representation of a unit number, room number, apartment number, suite number, level or floor number, building number, street number, a street name and a street type. A full list of street types and their abbreviations is available from [26]. The enumeration constraints presented in the model for street types are not intended to be a complete treatment, just indicative of the types of values that would be contained within this entity. We do not try to capture addresses such as the "corner of street x and street y". We take a stance similar to that presented in [26] that a number on either street x or street y will uniquely identify the location. The use of corner appears to be a term of convenience for the service requestor. It is feasible for the locative common name to include the reference to the corner. Street numbers within our model are a many to many relationship this allows us to capture a range of street numbers (e.g. 9 - 11). Street addresses may include zero or more street directory references, considering that the address may appear in street directories published by different providers.

We refer to street addresses as being in proximity to another street address. For example, a brochure for a local bowling alley states that they are "opposite the cinema". This additional locative information is intended to assist the requestor of the service. This proximity is captured in Figure 19 using a preposition. The enumeration constraints that we present in the model (such as opposite, next, above, below) are not intended to be complete but rather indicative of the type of information that the Preposition entity captures.

Postbox addresses we refer to as having either a general box number, a private box number or a locked bag number. They also inherit the same properties of the Address supertype that was mentioned above.

3.3.5 Street directory reference

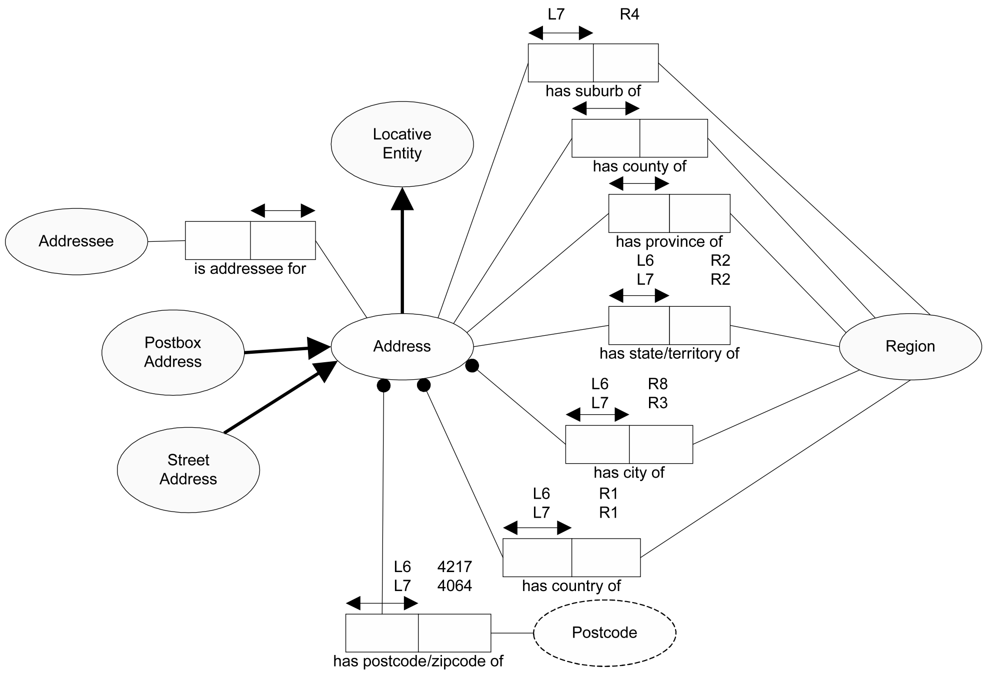
Street directory references are a common mechanism for service providers to provide assistance with physically arriving at a place of service provision [see Figure 21]. Street directories are commonly published books that include the following properties: an ISBN code (that uniquely identifies the book), the book title, the provider, the edition (e.g. 3rd edition), the region that the map relates to (e.g. Sunshine Coast or Gold Coast), the map number, the x and y position within the map, and the publication date.

The ReferencePosition entity depicted in the model has indicative enumeration constraints of 1 to 1000, and A through Z. This entity is used to reflect the X and Y position on the street directory. The values of the enumeration constraints are intended to be depictive only and should not be regarded as complete.

3.3.6 Phone numbers

Phone numbers are commonly included in the service descriptions as a means of requesting the service. We distinguish between fixed and mobile/cell phone types [see Figure 22]. All phone numbers are considered to be functionally constrained to support one or more of the following interaction types: voice, modem, SMS text messaging, facsimile and telex communications. We consider that it may be necessary to capture the international direct dialling prefix, the country code, the national direct dialling prefix, and the city/area code. It is mandatory to store the local phone number.

The discovery of information in this model is dependent on where you are calling from and where you are



each Address **is a** LocativeEntity **that** is of LocativeEntityType 'A'
each PostboxAddress **is a** Address **that** is of LocativeEntityType 'PA'
each StreetAddress **is a** Address **that** is of LocativeEntityType 'SA'

Figure 16: Addresses.

calling to. If you are located in Victoria (Australia) and you are calling someone in the state of Queensland then you need to ring the national direct dialling prefix "0", the area code "7" and the eight digit local number. If you are located overseas in Germany (and you wish to ring the same Australian local number) then you would need to ring the international direct dialling prefix "00", the country code "61", the area code "7" and the eight digit local number. The international direct dialling prefix will be dependent on the region that you are calling from, and possibly the telecommunications carrier that you are using to make the call. We support the notion of toll free (or free call) phone numbers for callers from certain regions.

The following example ConQuer query filters phone number instances to be those of "FixedLine" type that have a country code of "61" (being Australia) and an area code of "7" (for Queensland).

PhoneNumber
⊢ has phone number type "FixedLine"
⊢ has country code 61

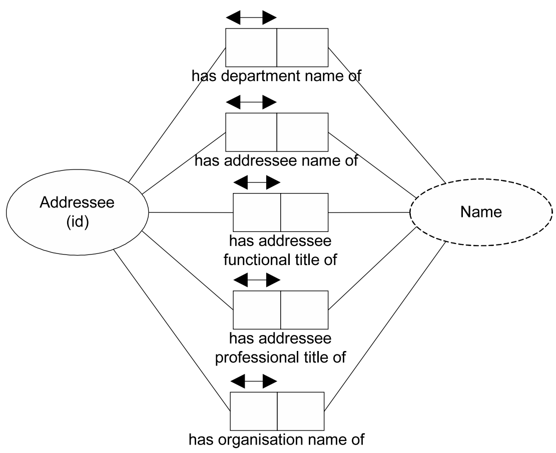


Figure 17: Addressees.

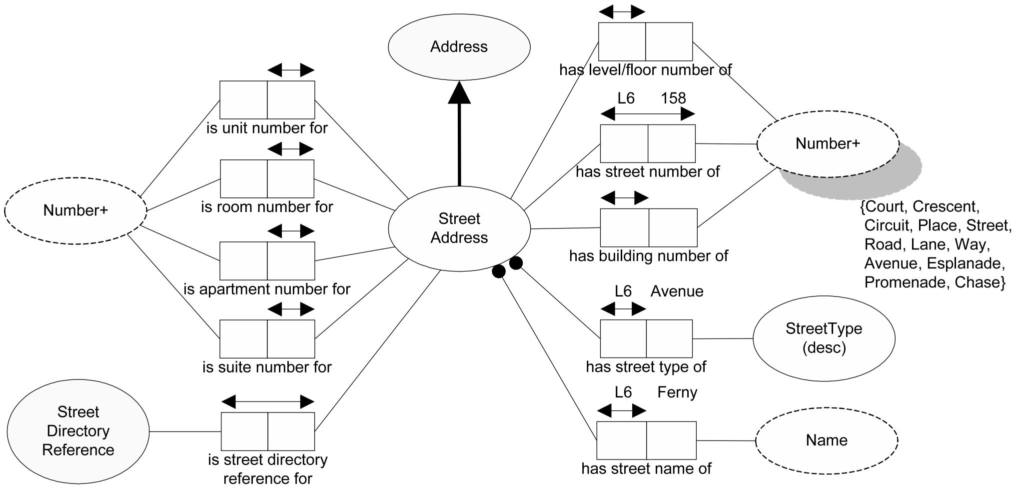


Figure 18: Street addresses.

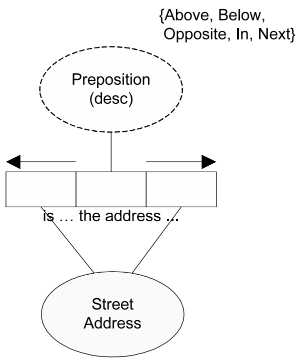


Figure 19: Proximity of standard addresses.

⊢ has an area code 7

3.3.7 Uniform resource identifiers

Uniform Resource Identifiers (URI) are used to describe abstract or physical resources [5]. They consist of a scheme type (e.g. ftp, http, mailto, news, gopher, telnet), an authority, a path and possibly a query [see Figure 23]. An authority represents the domain of the server. It includes a server name, userinfo and a password. Although the latter two are not recommended for use due to security concerns, we include them here for completeness. We consider web services to be capable of using URIs as a means of identifying the endpoint that facilitates the service provision.

The following example query filters for instances of URIs that have a scheme type of "http" and where the authority server name contains the word "google". The use of the contains function within the query enables us to get authorities such as gmail.google.com, labs.google.com and www.google.com.

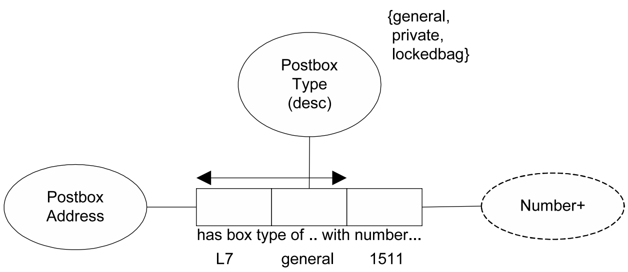


Figure 20: Postal box addresses.

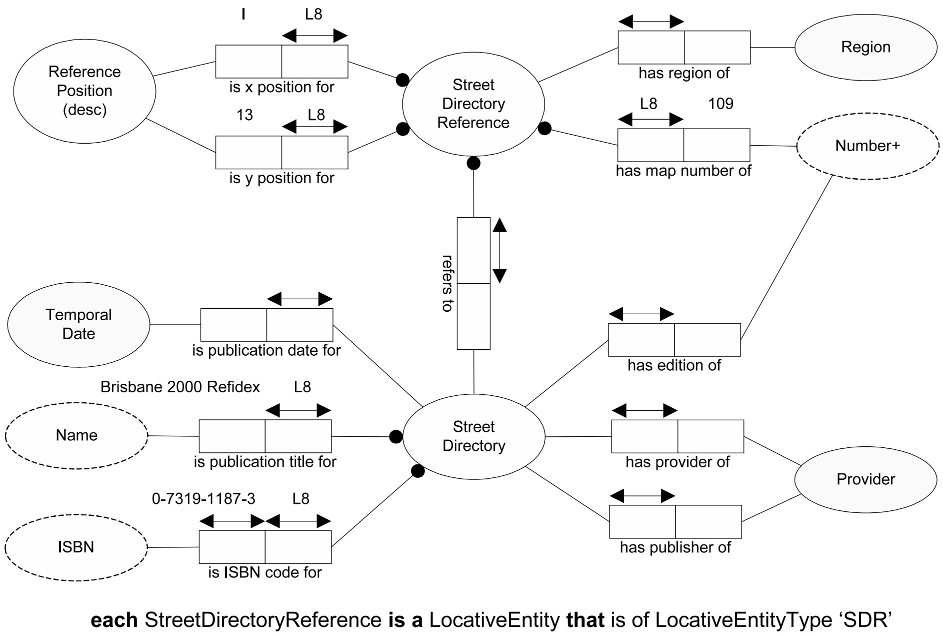


Figure 21: Street directories.

URI
⊢ has scheme type of "http"
⊢ has Authority
⊢ has server name like "%google%"

3.3.8 Internet protocol addresses

The Internet Protocol (IP) was designed to provide functionality for non-reliable, and non-sequential delivery of data packets (i.e. a collection of bits also known as a datagram) from a source address to a destination address. The transfer of these packets may involve many interconnected networks [1].

IP addresses are 32-bit, four part addresses used to define the network and host connected to an internet protocol network [see Figure 24]. IP addresses have a class A, B, C and local address components. Each part is referred to using a number between 0 and 255. The four parts are concatenated and are represented in the form 131.181.254.101 (registered to the Queensland University of Technology). Whilst a relationship exists between the IP address and a URI authority server name we believe that service providers will utilise one technique for description, not both.

We do not currently model IPv6 addresses.

3.3.9 Ethernet addresses

Ethernet addresses are used to uniquely describe a computer connected to an Ethernet network [see Figure 25] [10]. Ethernet addresses are a 48 bit address. They consist of two parts, the first is an identifier for the Ethernet hardware provider (also called an Organizationally Unique Identifier or OUI), the second is the address of the computer on the network. They are commonly written in hexadecimal format and it is the concatenation of the two parts that are used to uniquely identify an ethernet networked device. Ethernet address are sometimes referred to as physical, hardware or MAC addresses.

As the IEEE assigns the OUI, this effectively reserves a range of addresses for that provider. Whilst both parts of the address are 24 bits in length the true length of the first part is 22 bits. The other two bits denote a:

- ◇ Individual or Group address - Identifies an address as an individual or group address; and a
- ◇ Universally or locally administered address - Identifies whether the address has been assigned by a universal or local administrator.

Both of these can be derived by looking at the OUI. Using the individual or group bit provides for approximately 32 million (16 million local and 16 million group) addresses.

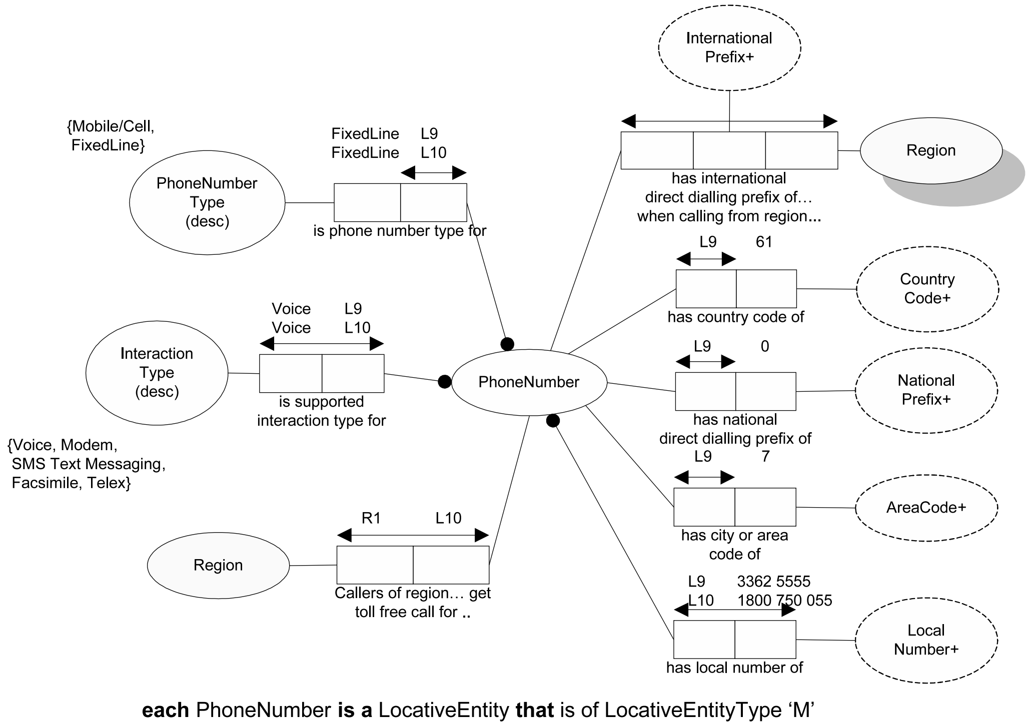


Figure 22: Phone numbers.

3.3.10 Spectra

Spectra is the collective term that we use to describe electromagnetic waves (or radio waves) for AM radio, FM radio, citizen's band (or CB) radio, TV, microwave, infrared, short wave, and radio frequency identification (RFID) that operate within specific frequency bands [see Figure 26]. For each spectra we capture the region within which the spectra is available. This caters for variations between countries for each type of spectrum.

We consider each spectrum to operate within a frequency band. A frequency band is defined by lower and upper boundaries for a specific band (e.g. AM radio normally operates in the 535 kilohertz to 1.7 megahertz range). When the lower and upper frequency are defined, they include a frequency number (e.g. 535) and the oscillation frequency units (e.g. kilohertz). The oscillation frequency refers to the number of cycles per second that the radio wave oscillates at. Therefore 535 kilohertz is 535 thousand cycles per second. It is not within the scope of this paper to present a full list of frequency bands by region.

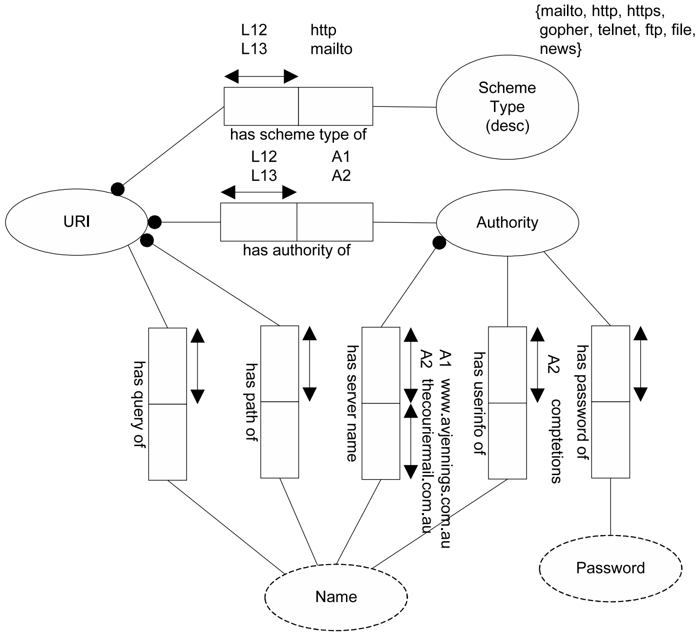
RFID tags can operate in either low, high, UHF or microwave frequencies [32]. We do not presently consider the fact that the RFID tag is either active or passive (referring to the manner in which the tag is powered), or the fact that it is assigned a globally unique identifier to be of high relevance for capturing the locative availability of a service.

Whilst other spectra exist for technologies such as garage door openers, baby monitors, traffic control systems and global positioning systems, we have tried to provide for the primary frequency bands that are in use. We do recognise that mobile phones operate using radio waves (typically in the 824 to 849 Mhz range) but chose to model them with phone numbers (see section 3.3.6).

3.4 Service availability

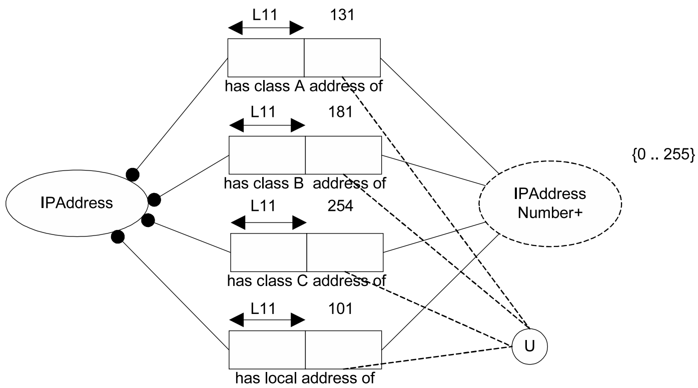
Service availability refers to the combined use of temporal and locative aspects of services to describe when they can be requested, provided, approached for issue resolution, approached with feedback, or queried for more information. We have split our modelling of service availability into two parts - availability related to the requests for services [see Figure 27] and availability related to provision [see Figure 28]. We deal with the former first.

Requests to providers can be for the service capability, to undertake issue resolution, to provide feedback or to gather information from the service provider. We consider the combination of the service, the request type (i.e. capability, issue resolution, feedback, information) and the locative entity to be the request locative availability [see Figure 27]. We apply the ability to filter (i.e. limit) availability for requests to be from specific locations. This is useful in the context of franchises that service locations such as regions (or a group of suburbs). Having shown where service requests can be conveyed to, we now allow for the attaching of the temporal availability of



each URI **is a** LocativeEntity **that** is of LocativeEntityType 'U'

Figure 23: Uniform resource identifiers.



each IPAddress **is a** LocativeEntity **that** is of LocativeEntityType 'I'

Figure 24: Internet Protocol addresses.

the request. We consider the requests to have three declared types of temporal availability:

- ◇ Nominated availability - The service requests are accepted at/during a specific temporal pattern (stated using a temporal entity).
- ◇ Negotiable availability - The service requests can be configured by the requestor in conjunction by discussing temporal availability with the provider.
- ◇ Continuous availability - The service is continuously available temporally for requests (e.g. for use in the context of web services).

Alternatively, a provider may elect not to advertise the temporal availability of requests for their service(s).

The provision side of availability is somewhat different as it offers the ability to describe either a specific locative entity that is "where" a service can be delivered to, or a type of locative entity (e.g. an address, a region, a URI etc). This is useful as a service provider may not want to provide a complete list of addresses that they are willing to provide a service to. In a manner similar to service request availability, we consider this locative aspect of provision to be referred to as provision locative availability. However, it is important to note that since the locative side of provision can be expressed in either of these two ways, the equivalent objectification of the locative fact type requires the introduction of the ProvisionLocation entity. We then allow for the attaching of three types of declared temporal availability with respect to provision: a nominated availability (i.e. a specific

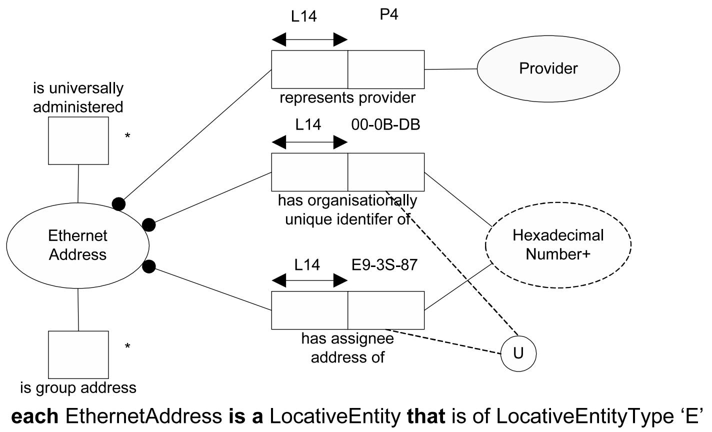


Figure 25: Ethernet addresses.

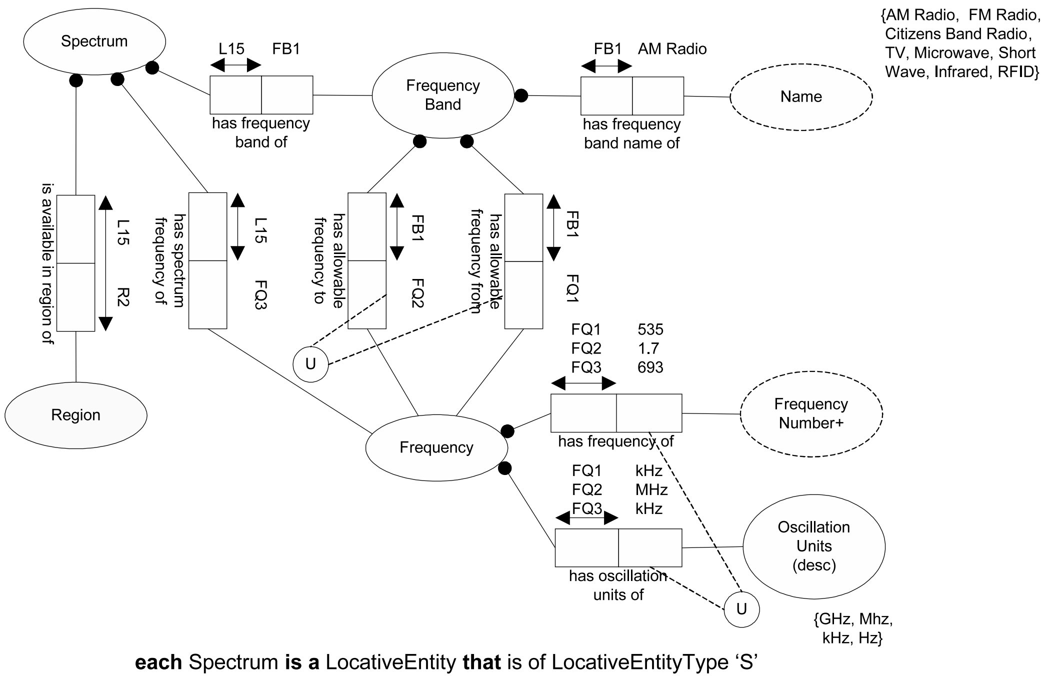


Figure 26: Spectra addresses.

temporal entity is provided), a negotiable availability or continuous availability. Alternatively, a provider may elect not to advertise the temporal availability for provision of their service(s).

3.5 Obligations

In order to capture the responsibilities of both the service provider and the service requestor we offer the notion of "obligations" as a means of ensuring that these non-functional properties are available for discovery by interested parties. We believe that by recording obligations, the discovery process will be greatly enhanced.

Obligations can be attached to either the request for a service, or the provision of a service [see Figure 29]. For example, a service provider may wish to advertise that service requestors have an obligation for a relationship, or an obligation to make payment should they request their service. It is the service provider who must fulfil the obligations relating to the service provision. As one or more providers may be involved in the provision of a service, we represent the obligations of the service provider separately to service requestors. We don't associate obligations with individual service requestors, as it is unreasonable to expect a service provider to outline all instances of service requestors to whom the obligations apply.

We now discuss three obligations: pricing, payment, and relationship obligations [see Figure 30]. In future, other obligations may be added to further increase the expressiveness of service descriptions.

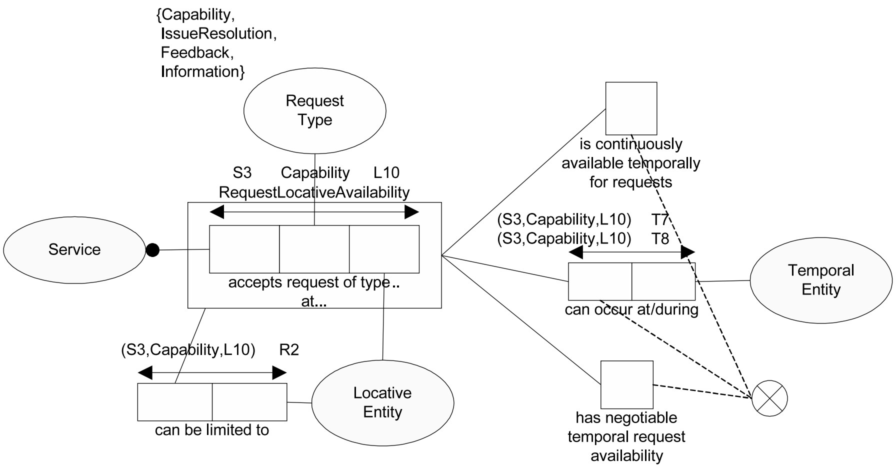


Figure 27: Service request availability.

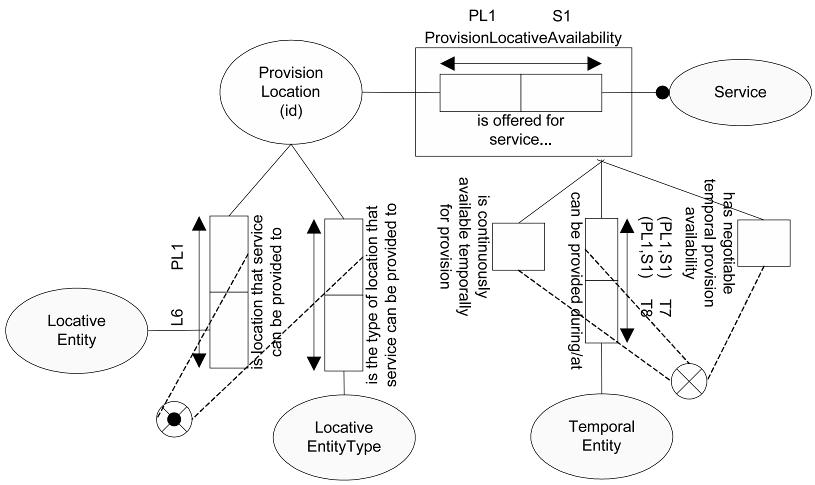


Figure 28: Service provision availability.

3.5.1 Pricing obligations

We view the need to price a service as an obligation of the service provider [see Figure 31]. The pricing obligation can be considered to wrap the price of a service with many other important non-functional properties. Most pricing obligations will primarily be concerned with capturing the price, but it is also necessary to capture information such as:

- Conditions - these relate to any specific requirements or restrictions to the price, or to the refund for the price paid for a service. Conditions are complex entities that we identify according to a URL. The constraints over the condition role allows for multiple URLs to be specified. This provides an ability to define fine grained conditions for use by service providers. Price conditions may include limiting service requestors to a particular group (e.g. elderly) or to the exclusion of a particular group (e.g. requestors of government agencies). Refund conditions are common for transportation services such as plane tickets where they state that a ticket may not be refundable, or may only be refunded within a particular timeframe.

- Refund procedure - associated with the specifying of refund conditions it may also be necessary for a service provider to state a refund procedure. This procedure is used by service requestors to enact the refund process. We consider procedures to be a sequence of steps that are followed to achieve an outcome. Our treatment of procedures is as per conditions. We utilise a URL reference to provide a link to the description of the procedure.

- Price validity - this provides a where and when scoping of the price's availability. Using our temporal models the temporal validity can be specified as an anchored or recurring interval, an instant or a date. We also capture the location as the pricing obligation may be specific to a limited number of the locations

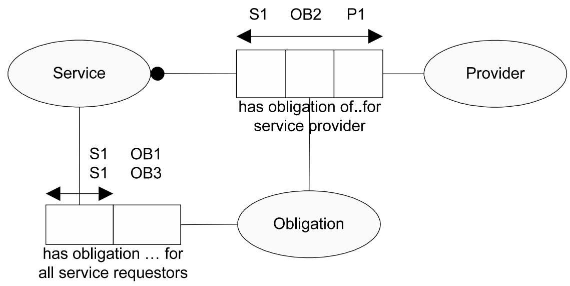


Figure 29: Service obligations.

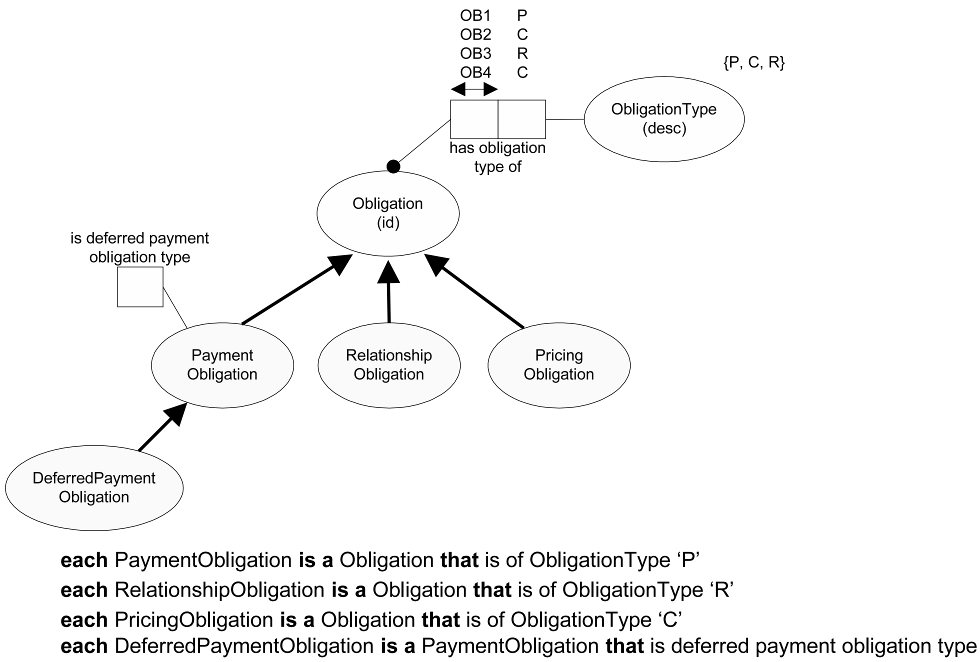


Figure 30: Obligations.

where a service can be requested from.

◇ Negotiability - sometimes the service provider may advertise a price but be willing to accept a lesser amount. Our model allows the provider to state that they are willing to negotiate on price.

◇ Price customisation - this allows the provider to capture that their service is highly customisable, and therefore the actual price cannot be expressed (e.g. a landscaping service may not be able to express the price until they have an understanding of the requestor's block of land and their objectives). This does not reduce the usefulness of the service description as the service provider is still capable of expressing the other pricing obligation related properties within this list.

◇ Relationship obligation - this allows the service provider to state that a relationship is required before they will commit to a price and its surrounding non-functional properties (e.g. conditions, discounts). See the outline of relationship obligations provided in the latter part of this section. It is possible to specify a relationship obligation within the model presented in Figure 29 that refers to the need to have a relationship with the service provider to receive the service output. We can also state a relationship obligation for the service requestor that is attached to a specific pricing obligation. For services that specify multiple pricing obligations (assumed to state different prices), then differing relationship obligations for those prices could be described.

◇ Payee discounts - we provide an in-depth discussion of discounts in section 3.8 but provide a link within our pricing obligation model to one specific type of discount, those related to who the payee is. This might include a person from a particular age group (e.g. the elderly), those with membership to a particular body, or even a shareholder of a company.

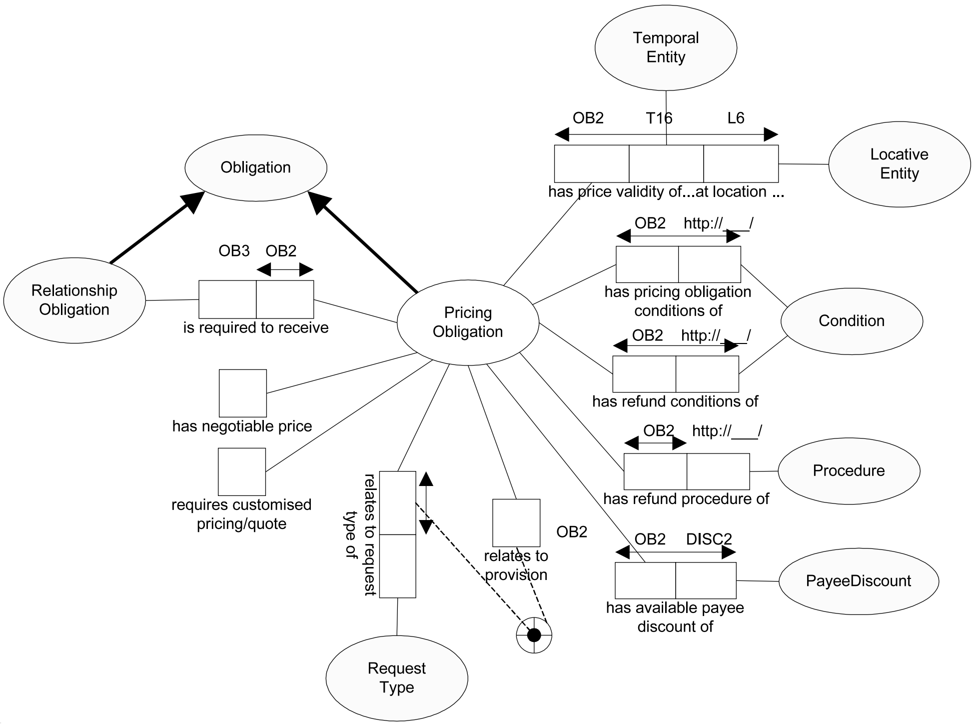


Figure 31: Pricing obligation.

As stated in our availability models, we view availability from the request and the provision perspective. Finally, we attach the price to either the type of request or to the provision. We leave it up to the service provider if they would like to state their price based on one of the request types (capability, issue resolution, feedback or information), or provision.

Deliberately, the actual price (e.g. $20 USD) of the service is not included within this model. We present a detailed discussion in section 3.6. This section was intended to be an overview of the notion of a pricing obligation.

3.5.2 Payment obligations

The payment obligation is the service requestor's obligation to pay the service provider. The payment obligation primarily provides a set of links to one or more pricing obligations. We have defined fact types within our payment obligation model [see Figure 32] for the following:

- ◇ Charge - this is the relationship from the payment obligation to the primary pricing obligation (i.e. the price of the service). We don't associate it directly with the price of a service as it is important to highlight the price of the service, as well as the other pricing related non-functional properties that we presented in section 3.5.1.
- ◇ Establishment fee - this allows the service provider to state that there is an obligation to pay a once-off fee for first time use of a service. Establishment fees (sometimes called "joining" fees) are common with gymnasiums and professional bodies.
- ◇ Interest charge - not specified by all services, interest charges are considered payable on services such as mortgage loans. If an interest charge is specified, it may be as an alternative to stating the charge of a service. An interest free period may be stated as a temporal duration.
- ◇ Administrative charge - this type of charge is common across many services. Examples include: (a) banks who charge monthly fees for some transaction accounts, and annual fees for some mortgages; and (b) rental property managers who charge monthly postage and petty expenses costs. An administrative charge is normally stated with a frequency (e.g. annual, weekly, monthly, daily etc).
- ◇ Deposit/Bond - service providers can choose to outline the need to provide a deposit or bond that accompanies a service. Deposits are common for a range of services including when people rent a property, or when borrowing hire equipment. This is another example of the usefulness of attaching the payment

obligation to the pricing obligation, rather than the price. In doing so, we can then navigate to the refund conditions and procedures for the deposit/bond.

- ◇ Conditions - we allow for the attachment of conditions to the payment obligation.
- ◇ Payment discounts - we provide a link within our payment obligation model to one specific type of discount, those related to how you pay. For example, this includes discounts related to the use of certain payment instruments, and payments to a certain type of location.

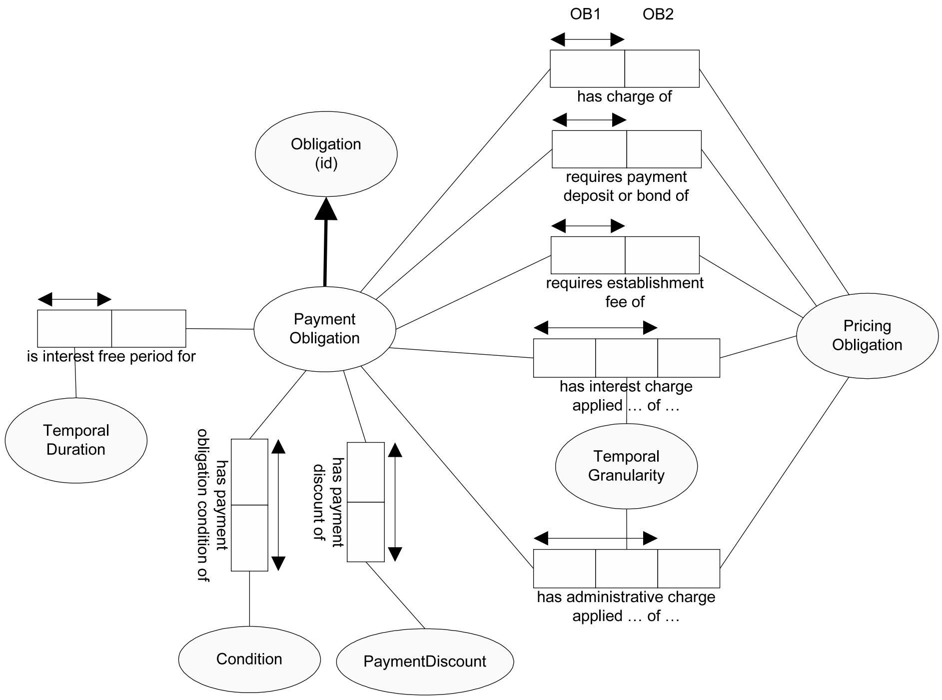


Figure 32: Payment obligation.

Deliberately, the manner in which a requestor can go about making payment (e.g. using a credit card) to the service provider is not included within this model. A detailed discussion of this and the properties of payment instruments is provided in section 3.7. This section was intended to be an overview of the notion of a payment obligation.

A subtype of payment obligation is "DeferredPaymentObligation". These are a specialisation of a payment obligation [see Figure 33] that enables a service provider to offer deferred payment for a service for a specific period (possibly after provision of the service has completed). Our deferred payment model includes support for specific deferred payment conditions, a deferred payment period stated as a temporal duration (e.g. 1 year), one or more temporal entities that represent a schedule of deferred payments, and the minimum amount that must be spent by the service requestor on the service to enable them to get the deferred payment option. All deferred payment obligations also inherit the properties of the supertype. Deferred payment obligations are common for retailers selling furniture, whitegoods, and entertainment systems. For example, large retailers may offer an 18 month (or more) interest free period for purchases over a certain amount (e.g. $500).

3.5.3 Relationship obligations

We provide the ability to capture an obligation that is a mandatory commitment by the service requestor to the service provider for a specified period [see Figure 34]. We refer to this as a relationship obligation. Relationship obligations include a set of conditions that govern the relationship, a minimum relationship duration (e.g. 12 months), and an associated set of rights. We allow a service provider to attach a set of rights that are associated with the relationship obligation. For example, this may be a service provider's right to disclosure, or a service requestor's right to privacy or recourse.

We have a specific subtype of relationship obligation that we call "Membership". Memberships are a common term used to describe an extended commitment. Memberships usually include a set of membership levels. We optionally attach a price to membership levels.

It is important to note that whilst each of the obligations outlined above has a specific fact type for capturing conditions, it is possible to have represented this as a single fact type associated with the Obligation supertype.

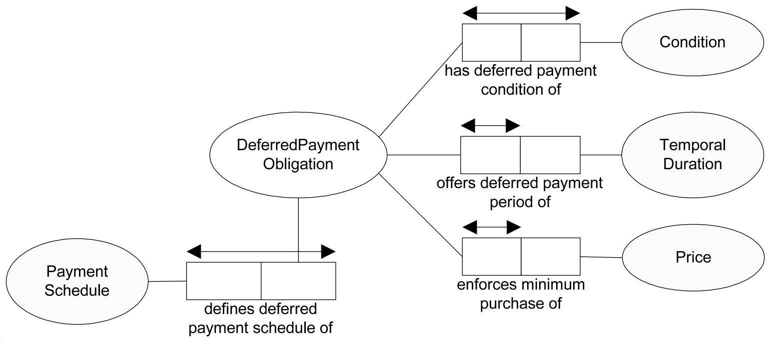


Figure 33: Deferred payment obligation.

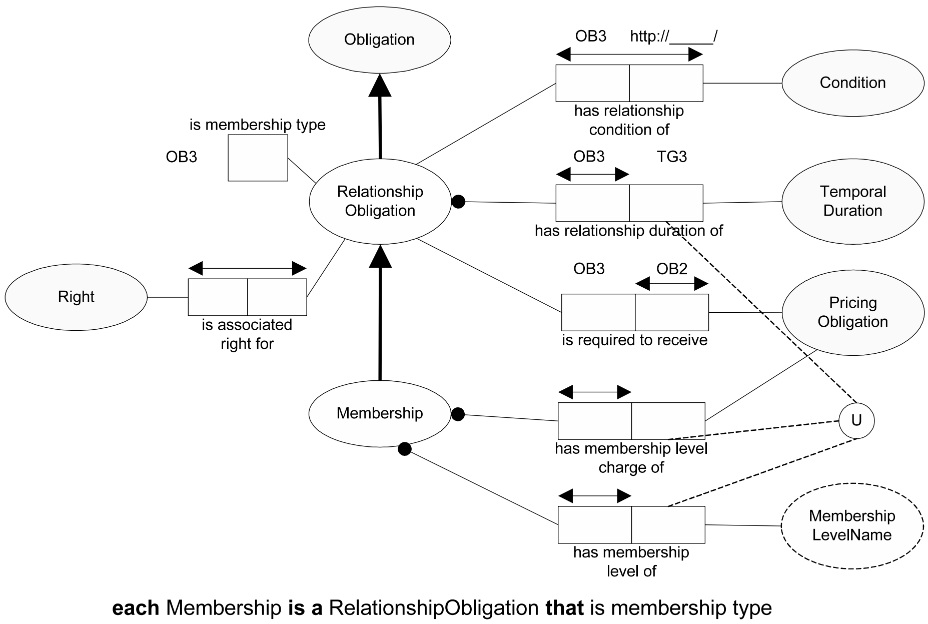


Figure 34: Relationship obligation.

We have chosen specifically to attach the condition to each subtype to retain further semantics about what the conditions relate to (i.e. deferred payment, relationship commitment).

3.6 Price

We perceive the non-functional properties of payment and price as complementary. We interchangeably refer to price as cost: cost being the view from a service requestor perspective, whilst price is the view from the service provider perspective. Within this paper we refer to price as the amount being charged for a service. We believe that the pricing of a service is an obligation of the service provider, one of many obligations involved with service request and provision.

Examples of price descriptions include:

- ◇ Carpet cleaning service - a carpet dry cleaning service offers 3 rooms cleaned for $89 AUD (where the maximum room size is 13 sq m, and subject to inspection of the carpet condition). They also offer 2 rooms for $69 AUD with additional rooms $25 AUD per room. Four rooms cost $110 AUD.
- ◇ Newspaper delivery service - a newspaper offers home delivery of newspapers daily for $7.20 AUD per week (i.e. 7 days for $7.20 AUD).
- ◇ Accommodation service - a hotel in Surfers Paradise is offering a room for $82.50 AUD per adult twin share.
- ◇ Mobile phone service - a phone carrier provides a mobile phone service. It is $10.38 AUD per month for 24 months for the handset, whilst the associated plan costs $25 AUD per month.

◇ Property investment syndicate - this investment vehicle offers 4.0% per annum income paid each six months. There are no fees.

From these examples we can see that prices are complex entities. They are not always easily captured as a simple dollar value in a certain currency. Prices do become quite domain specific when the quantifiers (e.g. per room) are applied. Certain complex conditions may also surround the ability of a service requestor to receive the advertised price.

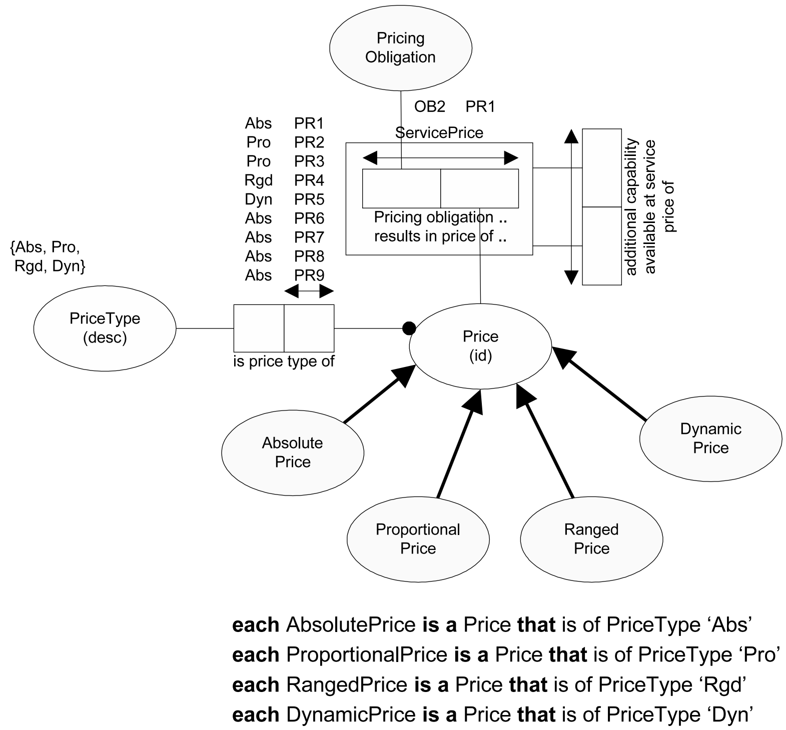


Figure 35: Price.

We consider that the pricing obligation of the service provider, in conjunction with the price, produces a new entity that we refer to as the "ServicePrice". We attach further information to this entity later in this section (e.g. tax, price granularity, price modifier). We also consider that once stating a price (e.g. 10 nights at $150 USD per night) the service provider might like to attach the price for additional invocations (e.g. each extra night is $100 USD per night). We assert that every price is one of the following [see Figure 35]:

◇ Absolute price - this contains a specific amount and a currency. For example $10 AUD represents ten (10) Australian dollars.

◇ Proportional price - this represents a percentage value with respect to a certain item. For example, the price of entering a managed fund might be 2.5% of the value being invested into the fund.

◇ Ranged price - Ranged prices are further subdivided into one of type types [see Figure 40]:

  – Ranged absolute - a ranged absolute price contains a *from* and *to* value that are both AbsolutePrice entities. For example, a service provider may prefer to provide a ranged price rather than a specific price (e.g. $150,000 USD - $175,000 USD). This means that the service provider may achieve a higher price if a service requestor is not familiar with the current market value of a service. In other cases, ranged prices reflect that various options are available that differ the final price of the service, depending on the option(s) that are selected.

  – Ranged proportional - a ranged proportional price contains a *from* and *to* value that both ProportionalPrice entities. For example, a service provider may state the cost of their service as a range between 1.5% and 3% of the final sale price.

◇ Dynamic price - this form of pricing captures mechanisms like auctions, where the price is determine by a market's natural supply and demand. We capture the type of mechanism (such as English auction, Dutch auction etc), the conditions associated with using the mechanism, the location and temporal availability of the mechanism, and a reserve price (as either an absolute or proportional price) [see Figure 41].

Price also includes an item granularity that is applicable to all types of prices [see Figure 36]. The item granularity reflects a general granularity, a cardinality and an item granularity number that provides ordering of granularities. The granularity of the item in turn refers to a unit of measure. We foresee the use of common granularities such as those presented in Table 2. These granularities could be extended further to support notions such as a room. This caters for services such as the carpet dry cleaning example that was presented at the start of this section.

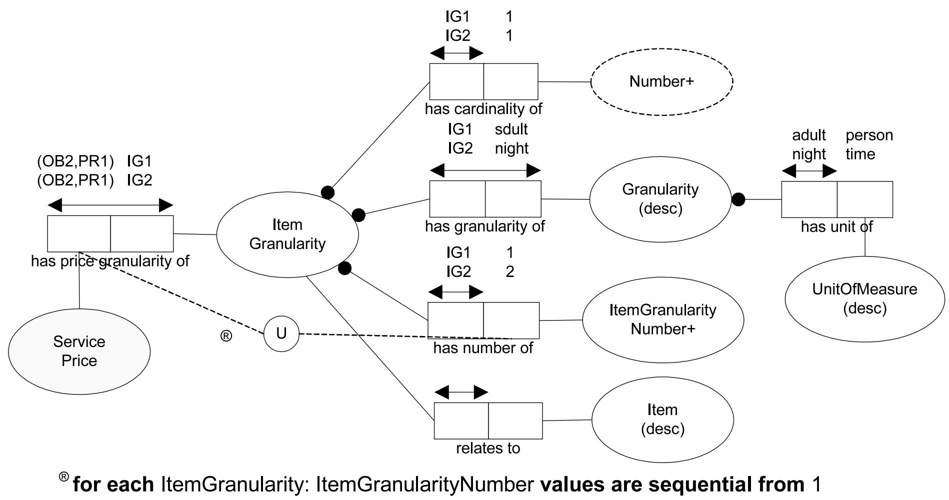


Figure 36: Price Granularity.

Unit	Granularity
Time	Hour, Minute, Second, Day, Month, Year, Night, Week, Fortnight.
Weight	Gram, Kilogram, Tonne.
Volume	Cubic metre.
Area	Metres squared, Square metres.
Length	Millimetre, Centimetre, Metre, Kilometre.
Watt	Kilowatt, Megawatt.
Byte	Kilobyte, Megabyte, Gigabyte.
Person	Adult, Child, Infant, Pensioner, Senior.
Event	Mouse click.
Permit	Ticket.

Table 2: Price granularity.

All prices have a modifier that quantifies the price being specified. We have provided four example modifiers: exact, limited to (the price will not go higher than the amount specified), inclusive (intended for ranges of values) and from (the price starts at this amount and will go higher depending on how the service is configured by the requestor). Prices may include a component that is tax related [see Figure 37]. Service providers can choose to state their price as inclusive or exclusive of a tax item. If a tax item is captured, then a tax percentage is attached. For example, Australians are taxed at a rate of 10% on the majority of goods and services they purchase under the Goods and Services Tax (GST). Similar taxes include the Value Added Tax (VAT). Tax is applicable to a particular region. Some services offer a price based on the criterion that the service requestor also requests the use of another service. An example is that the carpet cleaning service provider will offer their carpet protection service only when addition cleaning services are purchased. A service price may also provide either the service requestor or the service provider with one or more rights with respect to the service. Rights are outlined in more depth in section 3.10.

We provide a price matching facility within our price model [see Figure 38]. Some services advertise that they are willing to match or better the price of another competitor. For this type of service provider we allow the attachment of a percentage which indicates what they are willing to improve competitor offers by (e.g. 5%).

The following example query filters instances of price that are stated as an hourly rate between $10 and $15 United States Dollars (USD). This should not be confused with ranged prices where a service provider states the price using two absolute prices.

AbsolutePrice
⊢ has amount >= $10.00
⊢ has amount <= $15.00
⊢ has currency of "USD"
⊢ has price granularity of ItemGranularity
  ⊢ has granularity "Hour"
  ⊢ has unit of measure "Time"

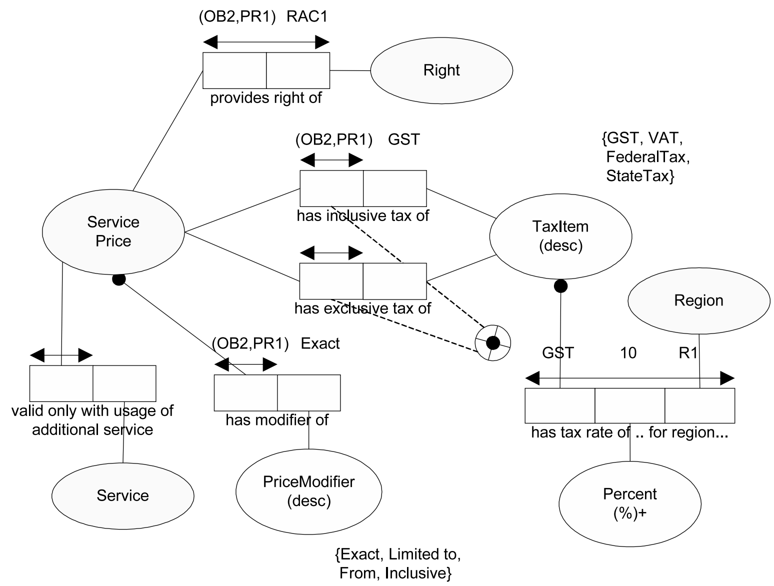


Figure 37: Pricing - Tax and modifiers.

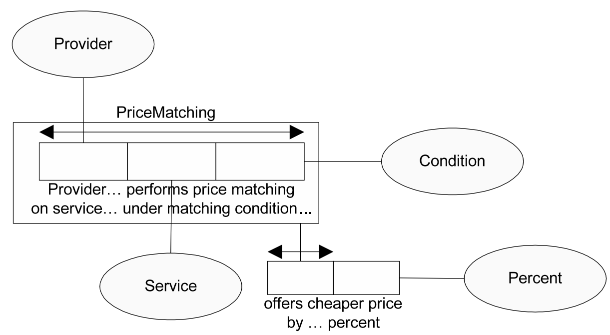


Figure 38: Pricing Matching.

⊢ has cardinality 1
⊢ has item granularity number 1

The following example query filters instances of price based on their annual, tax exclusive value where the proportional price ranges from 1.5% to 3.0% of “Funds under management”.

RangedProportional
⊢ has exclusive tax “VAT”
⊢ has price granularity of ItemGranularity
⊢ has granularity “Year”
⊢ has unit of measure “Time”
⊢ has cardinality 1
⊢ has item granularity number 1
⊢ has ranged proportional price of ProportionalPrice
⊢ has percentage 1.5%
⊢ has item “Funds under management”
⊢ has ranged proportional price of ProportionalPrice
⊢ has percentage 3.0%
⊢ has item “Funds under management”

Dynamic pricing mechanisms largely refer to online auction sites such as eBay where the market determines the price for a particular item. These mechanisms normally provide a temporal window at a particular location (e.g. a URL) where the mechanism is available. We use the TemporalInterval entity to allow for anchored

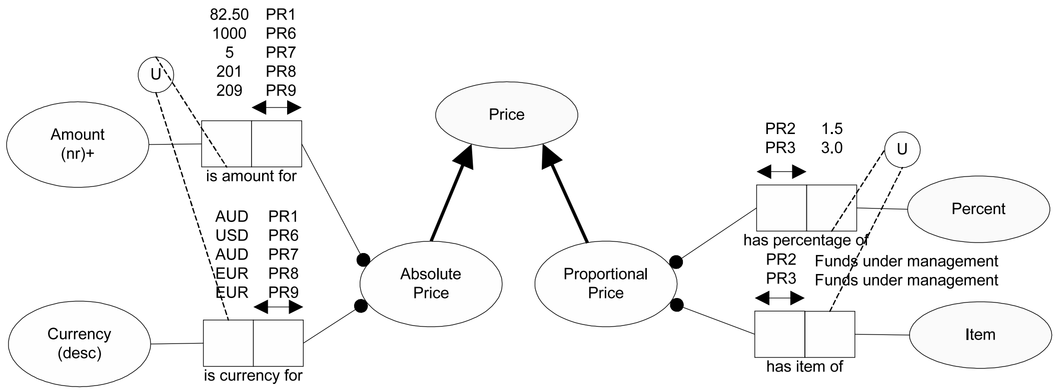


Figure 39: Absolute and proportional pricing.

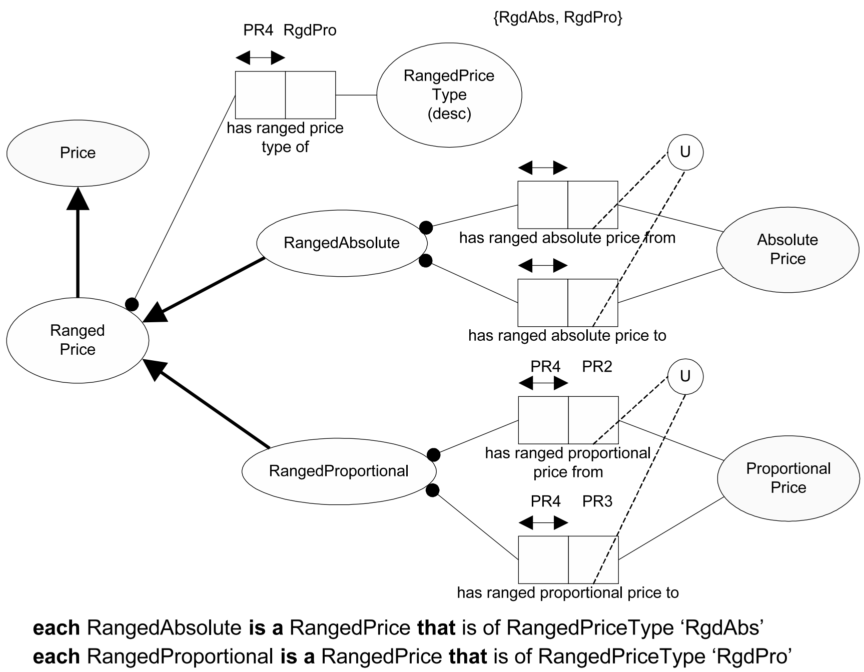


Figure 40: Ranged pricing.

intervals and recurring intervals. Auctions therefore can be described as occurring at a regular temporal interval (e.g. every Monday at 9am). Providers who are auctioning their services can state a reserve price and/or an indicative price. There are normally an array of specific conditions associated with these mechanisms. We do not attempt to capture these conditions. A mechanism type is attached to the dynamic price. This refers to dynamic pricing mechanisms such as Dutch auction, English auction, continuous double auction, Vickrey auction, sealed bids and request for tender. We provide a link to the provider of the dynamic pricing mechanism.

Some service providers choose to reward service requestors using loyalty schemes. We attach to the price of a service the possibility to accumulate rewards under a reward scheme [see Figure 42]. Reward schemes can be provided by the actual service provider or by a third-party service provider. Our model allows a service to attach a number of reward points to the invocation of the service, based on the service price that is paid (remembering that prices have a temporal and a locative availability). Reward points are only available during certain temporal intervals, or on a particular date, as well as being surrounded by some conditions.

3.7 Payment

In this section we present models for payment options, payment schedules, payment instruments and instrument types. We view payment as complementary to price. It captures the manner in which a service requestor can fulfil their payment obligation. We consider the following to be examples of payment related descriptions:

- ◇ A service has restaurant, catering and cooking school related capabilities. The service provider expects payment to be due on the invoicing date, or has 7 day terms for prior arrangement. It prefers the use of

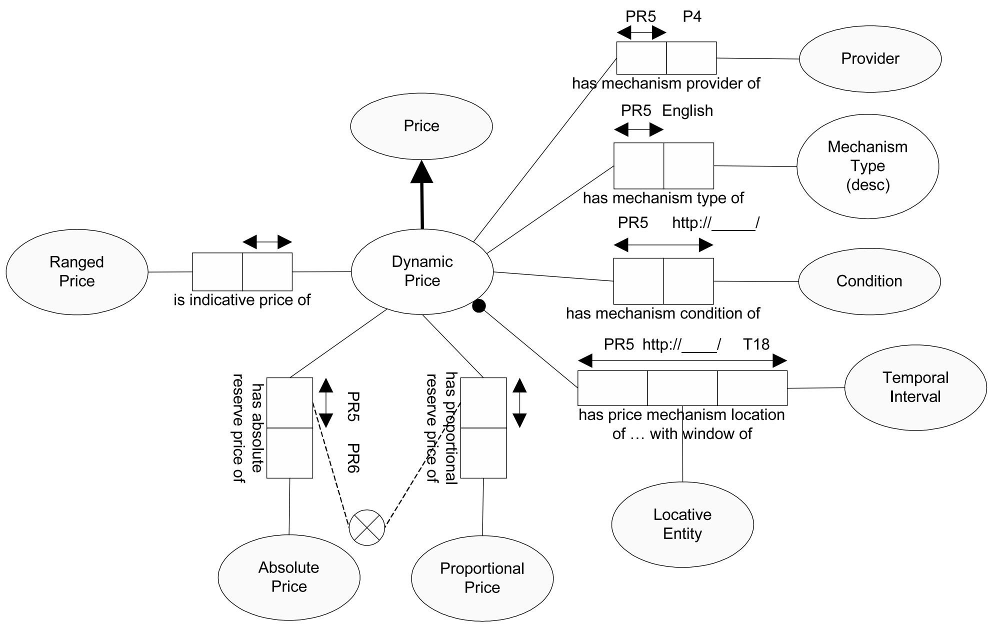


Figure 41: Dynamic pricing.

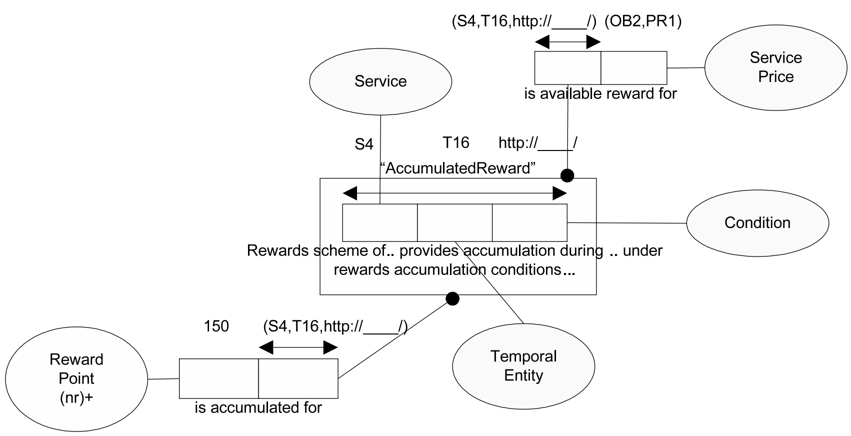


Figure 42: Service price rewards.

direct deposit or business cheque. Payment with an American Express or Diners card attracts a surcharge of 5%, whilst Visa, MasterCard and Bankcard attract a 2% surcharge. They offer a 5% discount for cash payment on functions over $1000 in value.

- An online auction service - the seller of a particular item offers payment options based on the purchaser's location. For requestors inside Australia the following are available: direct deposit/transfer, cheques, money orders, cash, PayPal or Paymate. For requestors outside Australia the following are available: International money orders, and Cash (Pounds sterling, Australian or United States dollars).
- An amusement park accepts Australian traveller's cheques, Bankcard, cash, bank cheques, Diners Club card, EFTPOS, foreign travellor's cheques, Mastercard and Visa.
- An electrician accepts EFTPOS, Mastercard, Visa, Bankcard, a personal or bank cheque and cash.

In simple terms we view a payment option as the preparedness of a service provider to accept a particular payment instrument(s) at one of several payment location(s) [see Figure 43]. The relationship between the payment obligation and a payment option is referred to as the "ServicePayment". The service payment primarily associates the use of certain payment instruments with a payment location (via the PaymentOption entity). It also identifies the following:

- Whether the payment option is a preferred payment option for the service provider.

- ◇ What the payment option terms of payment are from receipt of the invoice to the expectation for payment (stated as a temporal duration), and if the service also provides a tax invoice subsequent to invocation.
- ◇ Whether there is a charge associated with a particular payment option.
- ◇ Whether a payment option is only available to requestors from a particular region.
- ◇ Whether there are conditions that surround the use of the payment option; and
- ◇ When is the payment due? We capture a payment schedule with respect to a payment option in Figure 45. This schedule can be represented using one of two approaches. Firstly, the stating of a percentage of the overall price, and the temporal relationship to the service provision act (i.e. before, during, or after). Alternatively, the percentage can be stated with a temporal entity that applies. This allows for one or more temporal entities, which caters for the description of multiple payments.

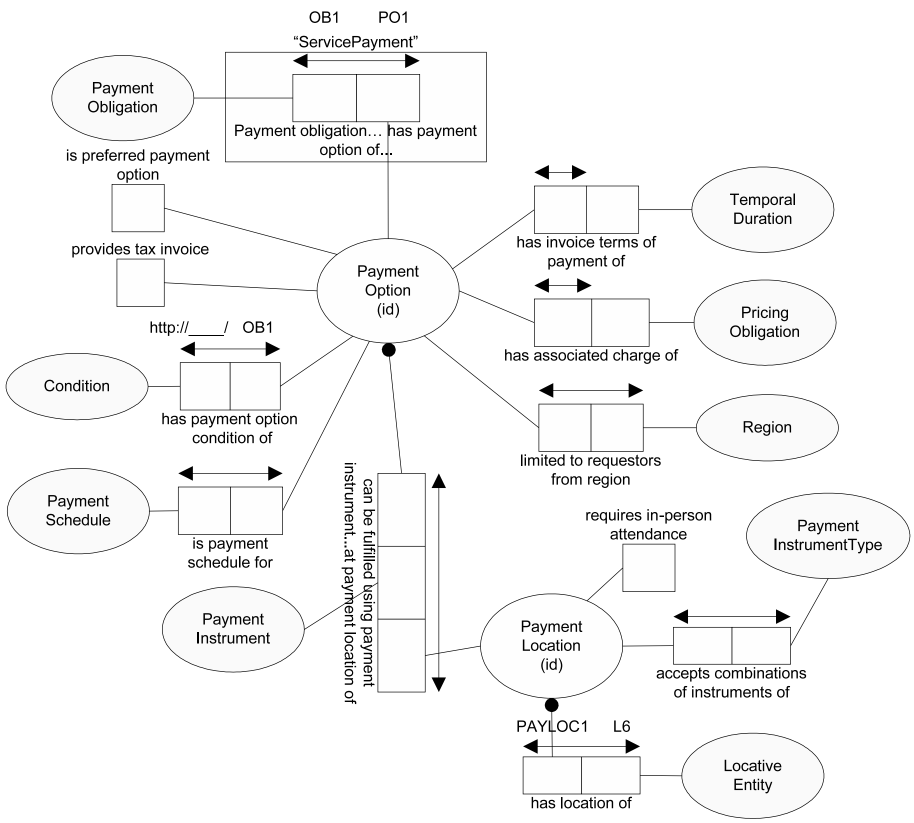


Figure 43: Payment options.

The following example query filters instances of payment options based on the following criteria: accepts credit cards with a terms of payment that expresses a temporal duration of 14 days.

PaymentOption
⊢ can be fulfilled using CardBasedInstrument
⊢ has card type "Credit"
⊢ has invoice terms of payment of TemporalDuration
⊢ has cardinality 14
⊢ has StandardTemporalGranularity
⊢ has standard granularity name "Day"

The payment location captures where the payment is to be fulfilled. This concept utilises the notion of locative entities to allow payments to be sent to multiple locations. We can group payment locations using their type from the locative model. This allows us to provide easier conceptual querying of payment locations. Examples of payment location types include postal addresses, phone numbers and URIs. This allows us to answer conceptual queries such as - Find me a service where the LocativeEntityType is "M" (referring to the

PhoneNumber subtype of LocativeEntity) and the payment location is an Australian phone number (+61)? This type of query may be useful for people that have access to a telephone but who do not want to call an international phone number to provide payment.

The following example query filters instances of payment options based on the following criteria: accepts instruments at a location that is a region with the common name of "Melbourne", and that issues tax invoices.

PaymentOption
⊢ has payment location of PaymentLocation
  ⊢ has location of Region
    ⊢ has common name of "Melbourne"
⊢ provides tax invoice

The ability to capture payment conditions provides the option to attach a condition to each of the payment options. For example, some providers want you to spend a minimum of $10 for all credit card transactions. Others restrict you from paying bills over a certain value via certain forms of payment instrument.

We view the redemption of rewards as a method for service payment in some circumstances [see Figure 44]. Reward schemes are an important way of maintaining customer loyalty and the redemption of such rewards is a sensitive issue. A number of reward points can be used during the redemption process. Like the accumulation of rewards, there are temporal constraints and associated conditions that control their use.

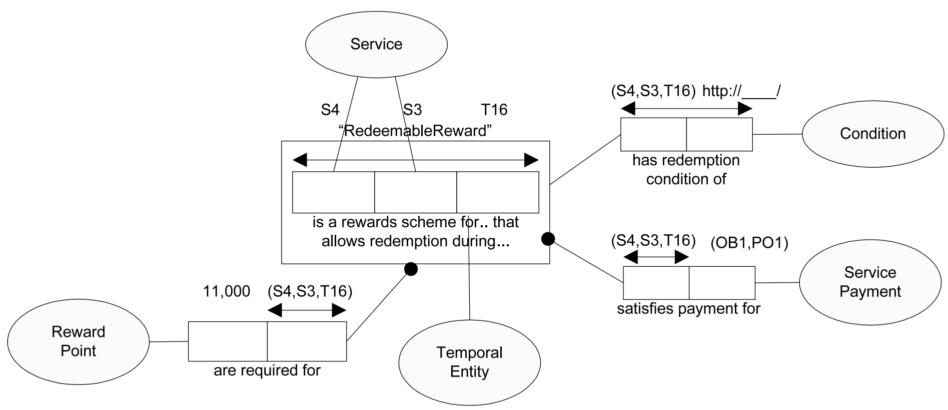


Figure 44: Service payment rewards.

Our models for payment instruments are presented in Figures 46 and 47. We consider payment instruments to have some common properties that are outlined in the first model. Payment instruments are issued by a provider. This includes cash, which is an instrument issued by a Reserve bank within a country. Payment instruments support one or more currencies and have associated locative information. This includes the region it was issued in, and the region(s) the instrument is limited to use in.

Attached to a payment instrument we allow a surcharge to be stated. This can be used to support the notion of additional costs for certain instruments, for example credit cards, where the service provider charges an additional amount when the service requestor uses the credit card. We consider some payment instruments to support payment schemes, which are in turn controlled by a provider. For example, a payment instrument such as a credit card might support the "Visa" payment scheme.

We consider that payment instruments have four subtypes. These categories are defined as card based instruments, cheques, cash, and voucher based instruments. Cards include credit cards (e.g. Visa, MasterCard), charge cards (e.g. American Express and Diners), debit cards, store cards and stored value cards. We group cheques into one of four types - personal, bank, business and travellers. We consider cash to be further subtyped into electronic cash instruments. This refers to electronic transfers such as direct transfer, direct debit, digital cash and wire transfers. Finally, we consider the payment instrument subtype of vouchers. We consider that vouchers should be stated as valid for redemption with a certain provider or service. They also have an associated temporal validity.

We expect that service requestors will be able to query for services based on the characteristics of payment instruments (e.g. traceability). A summary of payment instrument dimensions is available in [18]. These dimensions (referred to by us as characteristics) have been summarised here to reflect the complex issues associated with their representation. We envisage that service catalogues will facilitate this discovery based on payment instrument characteristics as a form of value-add.

The payment instrument characteristics [see Figure 48] include:

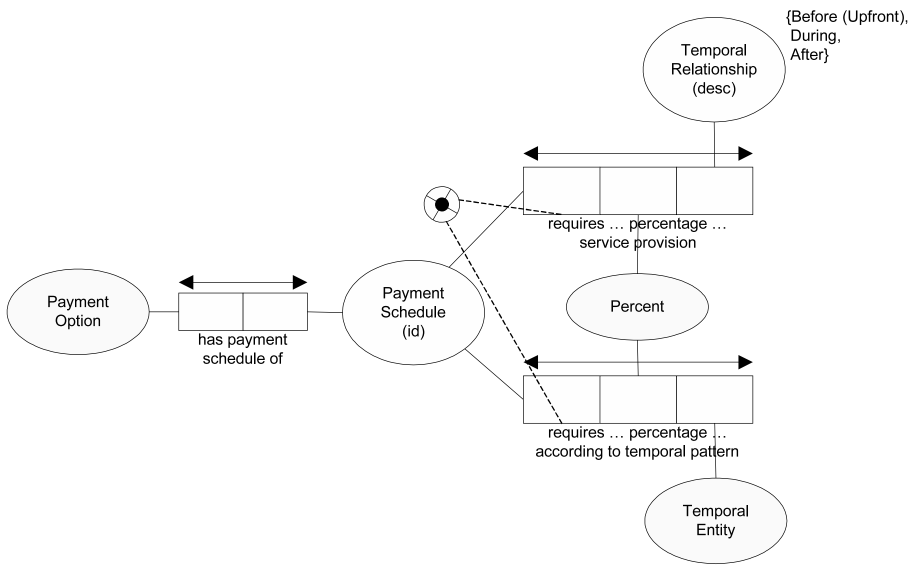


Figure 45: Payment schedule.

- ◇ Offline: Can the payment instrument be used in a non-electronic environment?
- ◇ Online: Can the payment instrument be used in an electronic environment?
- ◇ Acceptability: The relative acceptance of the payment instrument by the receiving party.
- ◇ Traceability: The service requestor, service provider (and any interim parties) and their associated operations/actions can be traced.
- ◇ Refutability: Neither party is capable of denying either payment or service receipt.
- ◇ Negotiability: Does the payment instrument have the ability to alter the negotiating conditions associated with the commodity or service?
- ◇ Liquidity: Is it possible to liquidate a holding in the payment instrument within a short timeframe?
- ◇ Expiration: Does the payment instrument have a fixed lifetime?
- ◇ Provider Coupling: To utilise the payment instrument are you coupled (e.g. by way of a card, account, password loyalty program or PIN) to the provider?
- ◇ Transferability: Refers to the ease with which an instrument can be transferred to another instrument.
- ◇ Security: Does the use of the payment instrument occur in a secure environment?
- ◇ Immediacy: How quickly is the value of the payment instrument transferred from one party to another?

Table 3 outlines the relationship between payment instruments and payment characteristics. Understandably subjective in nature, it should also be noted that this table will be different depending on the context within which it is viewed (e.g. countries like the United Kingdom have a high acceptance of cheque payments. This may not be the case in non-cash centric economies such as Japan). We consider the populations within the table to be indicative of the values that catalogues could provide to assist with discovery based on payment instrument characteristics.

3.8 Discounts

Discounting is a common method of attracting custom. Various types of discounts are available for services. We have previously shown in section 3.5 that pricing obligations and payment obligations may both attach their related discounts. We view discounts from the perspective of the service requestor, and therefore we believe that discounts can be categorised according to how you pay (e.g. early payment, coupon used), as well as to who you are (e.g. an elderly person) [see Figure 49]. We refer to this distinction as payment related discounts and payee related discounts respectively.

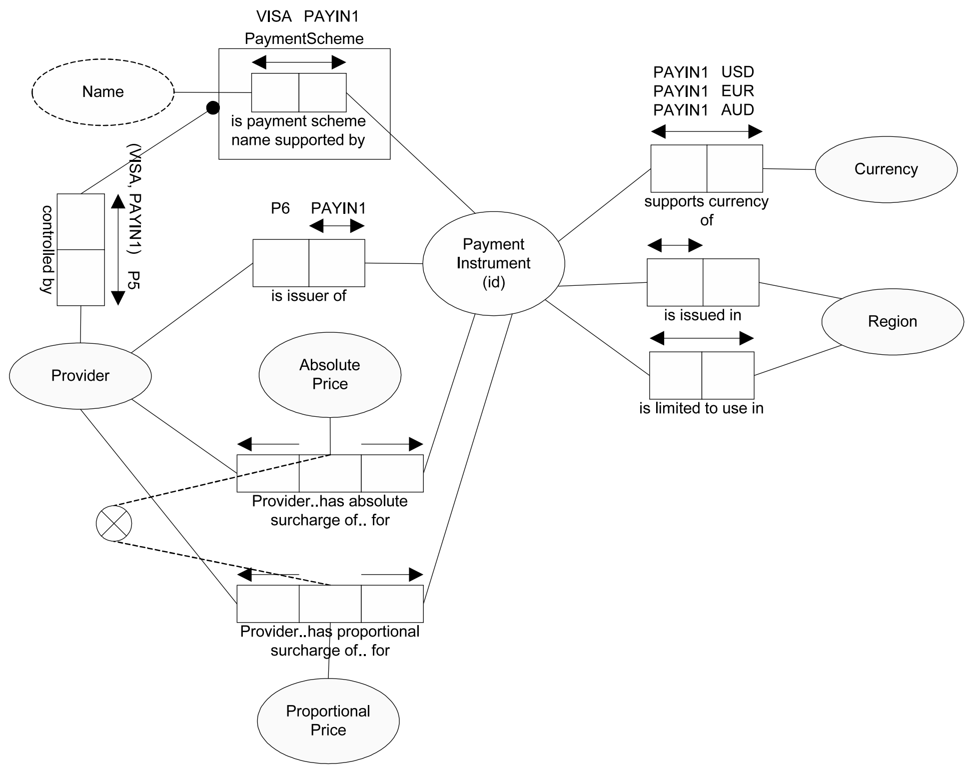


Figure 46: Payment instruments.

The service provider is unable to determine in advance all the combinations of service discounts that might apply to a price based on attributes of the service requestor (e.g. their age, membership to associations). For this reason, the catalogue provider (who may have more context related information with respect to the service requestor) may apply the discounts to a price before it is presented to the requestor during the discovery process. We therefore consider our notion of discounts to not be included within the price specified.

We consider that a service provider might want to state the discount in one or two ways: as a reduction of the price of a service, or as a resulting price for a service. In either case, it may also be a different service where the discount is available. This allows a service provider to entice a service requestor with a discount on the price of a service, and to then provide a discount on another (possibly more expensive) service of the provider. As we use the Price supertype, it allows for the inclusion of a range of possible discounts expressed using the RangedAbsolute or RangedProportional entities. For both payment and payee related discounts we attached an availability that expresses the temporal and locative constraints for the discount. Service providers may also state any conditions associated with receiving a particular discount.

We consider that service providers who offer their services for a cheaper price during a specified period (i.e. a sale) are capable of expressing this fact using the temporal validity aspect of our price obligation model. The price validity may refer to a date range when the price is cheaper.

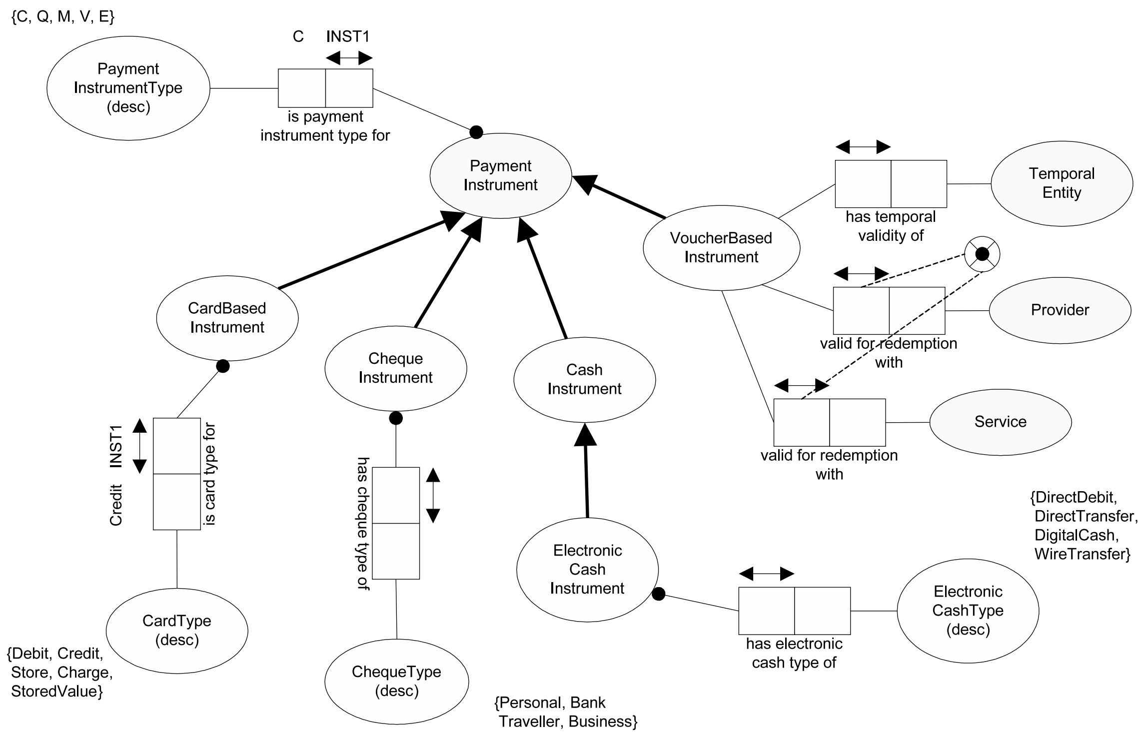
All payment related discounts optionally require a minimum purchase amount before the discount can be received. The following are payment related discounts [see Figure 50]:

- ◇ Early payment - this type of discount is offered to service requestors who are able to meet a payment obligation earlier than required. An early payment offset, that is a once-off period for receipt of the early payment, or an early payment schedule is captured.

- ◇ Type of payment instrument - this discount is provided based on the type of payment instrument being accepted by the service provider. For example, a service provider may offer a discount for payment using "Cash" as they are not incurring merchant fees associated with payment instruments like credit cards.

- ◇ Coupon/Offer - coupons are common paper-based mechanisms for advertising discounts for services. They are regularly constrained to a specific temporal pattern (eg. Valid until a certain date, valid between certain dates). This type of payment discount does not include gift vouchers as we consider these to be a form of payment instrument .

- ◇ Payment location type - this discount is provided based on the type of payment location that is used by the service requestor.



each CardBasedInstrument **is a** PaymentInstrument **that** is of PaymentInstrumentType ‘C’
each ChequeInstrument **is a** PaymentInstrument **that** is of PaymentInstrumentType ‘Q’
each CashInstrument **is a** PaymentInstrument **that** is of PaymentInstrumentType ‘M’
each ElectronicCashInstrument **is a** PaymentInstrument **that** is of PaymentInstrumentType ‘E’
each VoucherBasedInstrument **is a** PaymentInstrument **that** is of PaymentInstrumentType ‘V’

Figure 47: Payment instrument type hierarchy.

The following example query filters instances of payment discounts based on the following criteria: the price discount is an absolute price type, is a maximum of $50 Australian, and is for early payment at least 14 days in advance.

EarlyPaymentDiscount
  ⊢ has discount amount of AbsolutePrice
    ⊢ has amount <= 50.00
    ⊢ has currency “AUD”
  ⊢ has DiscountType “PaymentDiscount”
  ⊢ has early payment offset of TemporalDuration
    ⊢ has cardinality >= 14
    ⊢ has StandardTemporalGranularity of
      ⊢ has standard granularity name “Day”

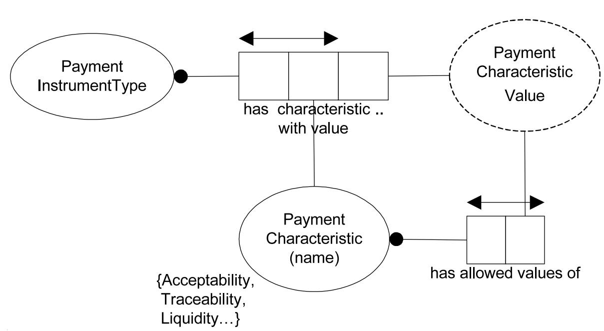


Figure 48: Payment instrument types.

Payment Instruments	Payment Characteristics											
	Offline	Online	Acceptability	Traceability [2]	Non-refutable	Negotiability	Liquidity	Expiration	Provider Coupling [3]	Transferability	Security	Immediacy
Cash	◇ [1]		H	◇		◇	H			◇		I
Cheque	◇		H	◇		P [4]	M	◇	◇	◇		D
Direct Funds Transfer		◇	H	◇	◇	◇			◇	◇	◇	I
Credit/Charge Card	◇	◇	H	◇			◇		◇	◇		D
Traveller's Cheque	◇		M	◇	◇	◇	M			◇		D
Wire Transfer	◇		L	◇	◇		M					D
Money/Postal Order	◇		L	◇	◇		M					D
Security	◇	◇	H	◇	◇		M		P			D
Bond	◇	◇	L	◇			H			◇		D
Bank Bill	◇	◇	L	◇			H			◇		D
Voucher	◇		M	◇		P	H	P	◇			I
Stored Value Card	◇	◇	L	◇			L	P	◇			I
Digital Cash		◇	L	◇	◇		L		◇	◇	P	I
Anonymous Digital Cash		◇	L		◇		L		◇	◇	P	I

Table 3: Characteristics of payment instruments.

[1] Subjective Legend : P = Possibly, H = High, M = Medium, L = Low, I = Immediate and D = Delayed. ◇ indicates the dimension is applicable to the payment mechanism.
[2] Traceability has varying degrees. Face-to-face cash transactions have a degree of tracability. However this type of transaction could also be conducted on behalf of someone else (e.g. by giving money to someone to buy something for you).
[3] Coupling includes items such as accounts, passwords, and personal identification numbers.
[4] Cheques can be marked "not negotiable".

As stated, payee discounts relate to who the service requestor is. The following are payee related discounts [see Figure 51]:

- ◇ Age group - service requestors who belong to specific age groups often find that they receive a discount. This is common for pensioners/seniors, infants, and children.
- ◇ Student - discounts are sometimes offered based on the requestor being a student. We provide a differentiation between school and full-time university students.
- ◇ Membership of a particular service sometimes brings with it discounts with another service provider. Large organisations (e.g. health funds) whilst requiring membership themselves, normally negotiate discounts with other service providers on behalf of their members.
- ◇ Shareholder - discounts are sometimes provided to shareholders. There is normally a requirement for holding a minimum number of units before the discount is available.

3.9 Penalties

Penalties are a mechanism for service providers to describe what will occur in the event that a service requestor does not comply with a specific obligation. Penalties are commonly outlined in service level agreements as a means of compensating the service requestor for non-performance (in the generic use of the term performance). An example of a condition under which a penalty is applied is for non-payment or late payment by the service requestor. Penalties will normally have a related set of conditions.

Figure 52 presents our model for penalties. We provide specific subtypes for four types of penalties: termination, financial, involuntary suspension and loss of right penalties. By termination we refer to the service provider ceasing to provide to the service requestor the output of the service. Termination is non-reversible. Our model for penalty introduces a link between the TerminationPenalty entity and a particular form of right

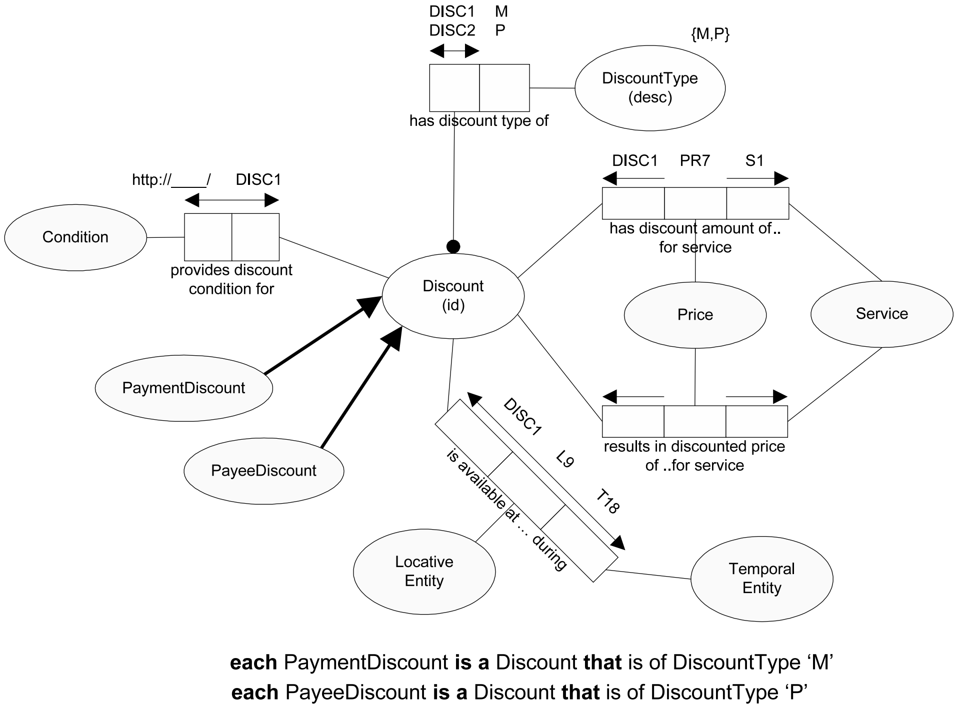


Figure 49: Discounts.

(rights are discussed in further depth in section 3.10) that we refer to as the RightOfTermination. This right of termination ensures that the provider does not have to honour their previous agreement with the service requestor.

A financial penalty is self-explanatory. Under this scenario, the service provider chooses to impose a financial punishment on the service requestor. This is represented in our model as a PricingObligation that is defined by the service provider. Involuntary suspension refers to the service provider's decision to temporarily interrupt the provision of a service to a particular requestor. Involuntary suspension within our model is represented as a link to the RightOfSuspension entity that is outlined later on in this document. For the moment it is sufficient to say that involuntary suspension is for a specified period, and has conditions, procedures and/or obligations surrounding the suspension and resumption.

Finally, penalties may result in the loss of one or more rights. This type of penalty can be used to capture the loss of warranty rights, the loss of access to a service, the loss of the right to recourse etc. Although, the involuntary suspension and termination penalties are linked to their associated rights, we have chosen to represent them as distinct subtypes of penalty to achieve greater discovery capability.

The following example query filters instances of penalties based on the following criteria: the penalty is imposed for not complying with a payment obligation, is an involuntary suspension that has a maximum suspension period of 10 days, and has an obligation on resuming that is a payment obligation for not more than 50 Euros.

```
InvoluntarySuspensionPenalty
  ⊢ is imposed for non-compliance to PaymentObligation
  ⊢ invokes service provider suspension of RightOfSuspension
    ⊢ has maximum suspension period of TemporalDuration
      ⊢ has cardinality <= 10
      ⊢ has StandardTemporalGranularity of
        ⊢ has standard granularity name "Day"
    ⊢ has resumption obligation of PaymentObligation
      ⊢ has base charge of PricingObligation
        ⊢ has price of AbsolutePrice
          ⊢ has amount <= 50.00
          ⊢ has currency "EUR"
```

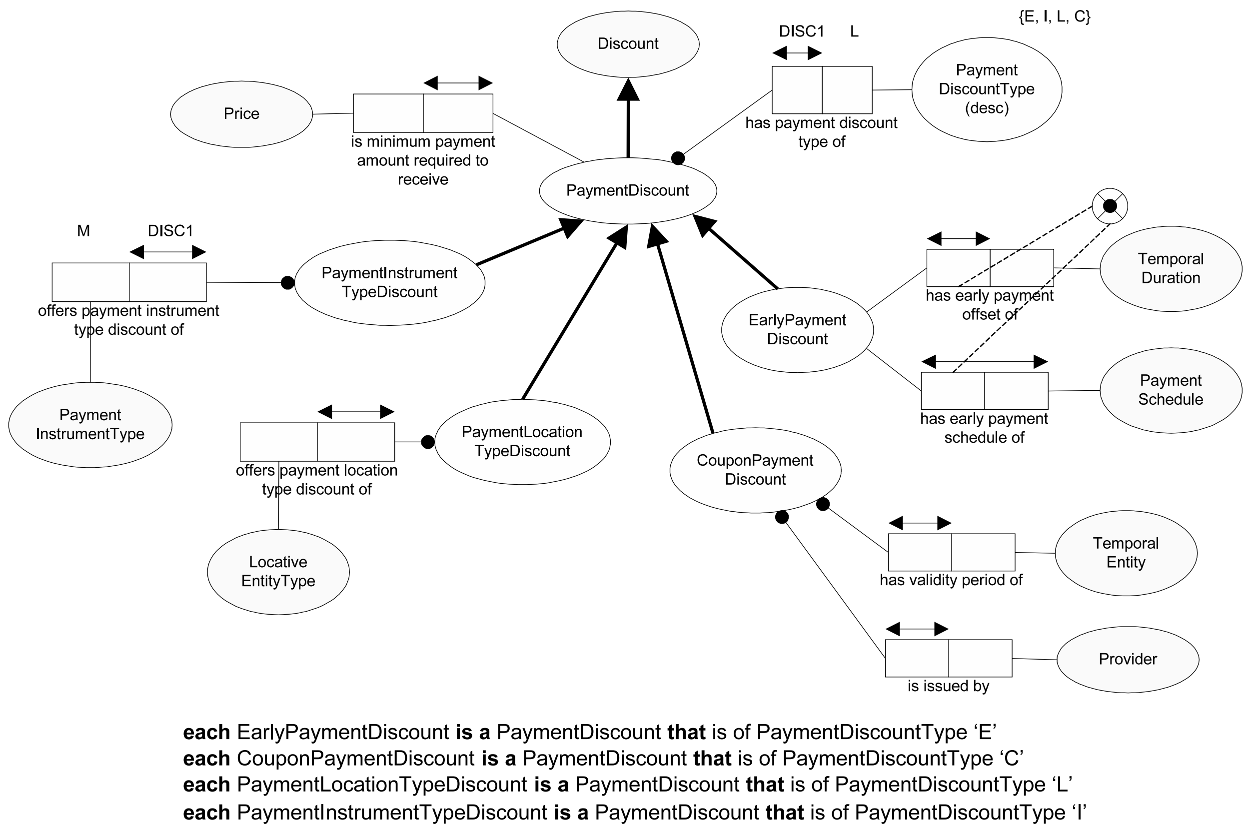


Figure 50: Payment discounts.

3.10 Rights

Rights are permissions granted to either the service provider or the service requestor in the environment within which the service operates. We have previously shown that rights can be used in two contexts. Firstly, that along with a service price there may be associated rights available to the requestor or the provider. Secondly, that the penalties for not meeting the obligations associated with a service can result in the loss of one or more rights of the service.

Rights include the following: access to service resources, recourse (or appeal), suspension/resumption, termination, privacy, warranty (or guarantee), refusal of service, disclosure, cooling off periods, liability limitation and extension to the original service provision commitment [see Figure 53]. We attach a name, a period of validity and a designator that outlines whether the rights are for the service provider or the service requestor.

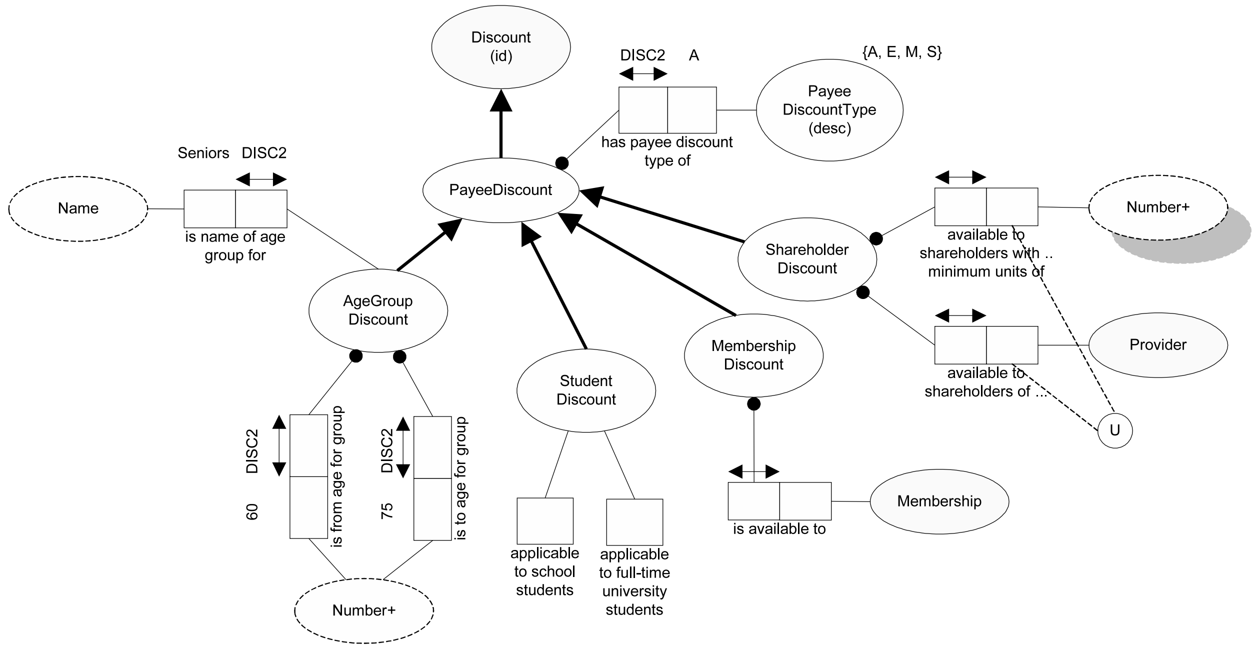
3.10.1 Access

Access refers to the right to the use of the service, normally for a specific period [see Figure 54]. This temporal validity is inherited from the "Right" supertype presented in Figure 53. The right is modified to include an access type. Access types describe a continuum of permission to use a resource and include exclusive, restricted, shared, and prohibited access. Exclusive access is normally only available when the service requestor incurs an obligation (e.g. an additional payment, or an extended relationship commitment between both parties) that is additional to the standard obligations for a service offering.

Access is provided to the service requestor over a resource that we identify generically according to a name and a type. Resource types include intellectual property, information/knowledge, a design (e.g. a registered trademark), person(s), a facility, and time. These are intended to be indicative of the enumeration constraints for the ResourceType entity. The access to these resources is normally provided for a specific temporal duration that incurs one or more obligations. We optionally provide the ability to store a location of the resource that the access is provided to.

3.10.2 Recourse

Requestors of a service have the right to recourse, or an appeals process [see Figure 55]. Recourse is usually available for a certain period of time after the service provision has been completed. Within this period, the

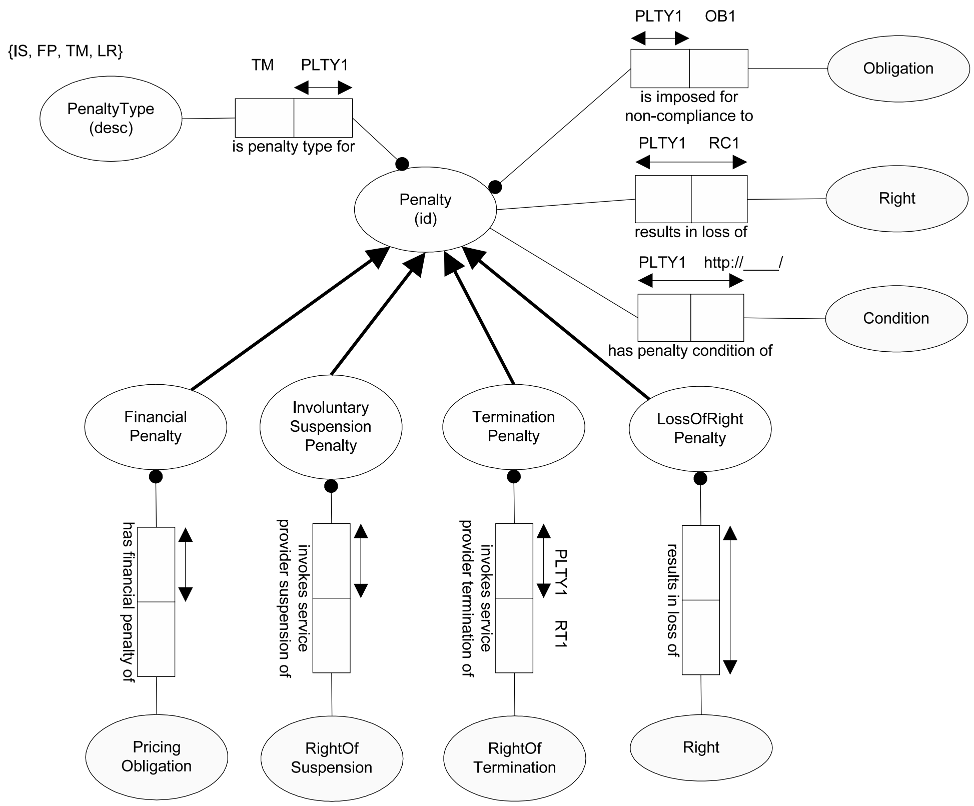


each AgeGroupDiscount **is a** PayeeDiscount **that** is of PayeeDiscountType ‘A’
each StudentDiscount **is a** PayeeDiscount **that** is of PayeeDiscountType ‘E’
each MembershipDiscount **is a** PayeeDiscount **that** is of PayeeDiscountType ‘M’
each ShareholderDiscount **is a** PayeeDiscount **that** is of PayeeDiscountType ‘S’

Figure 51: Payee discounts.

service requestor can utilise an appeals procedure in an attempt to rectify the matter at issue. Recourse is sometimes mediated through a separate third party provider. In the case where it is not mediated it is assumed that the service provider controls the process of recourse.

In a service environment, recourse is administered according to the laws of a jurisdiction. The jurisdiction refers to a governed locative entity (e.g. a state within a country) and legislation within that governed locative entity. Each piece of legislation is referred to according to an assigned name and a year of introduction. We also capture that legislation is superseded or amended by one or more pieces of legislation.



each InvoluntarySuspensionPenalty **is a** Penalty **that** is of PenaltyType 'IS'
each FinancialPenalty **is a** Penalty **that** is of PenaltyType 'FP'
each TerminationPenalty **is a** Penalty **that** is of PenaltyType 'TM'
each LossOfRightPenalty **is a** Penalty **that** is of PenaltyType 'LR'

Figure 52: Penalties.

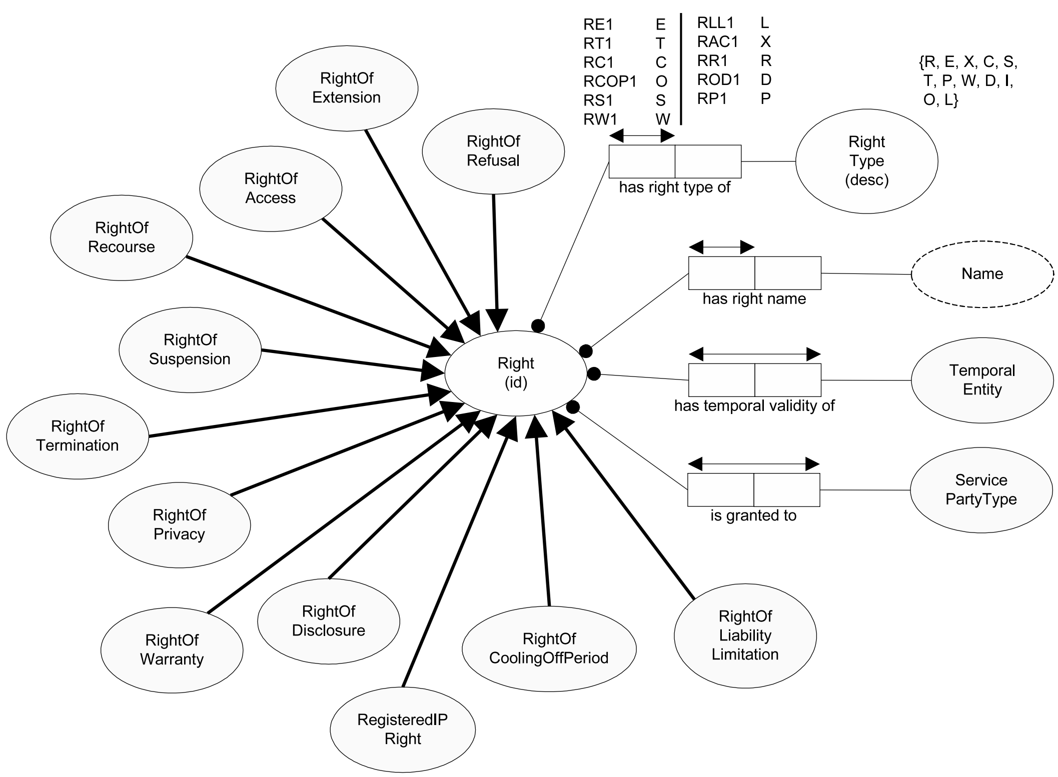
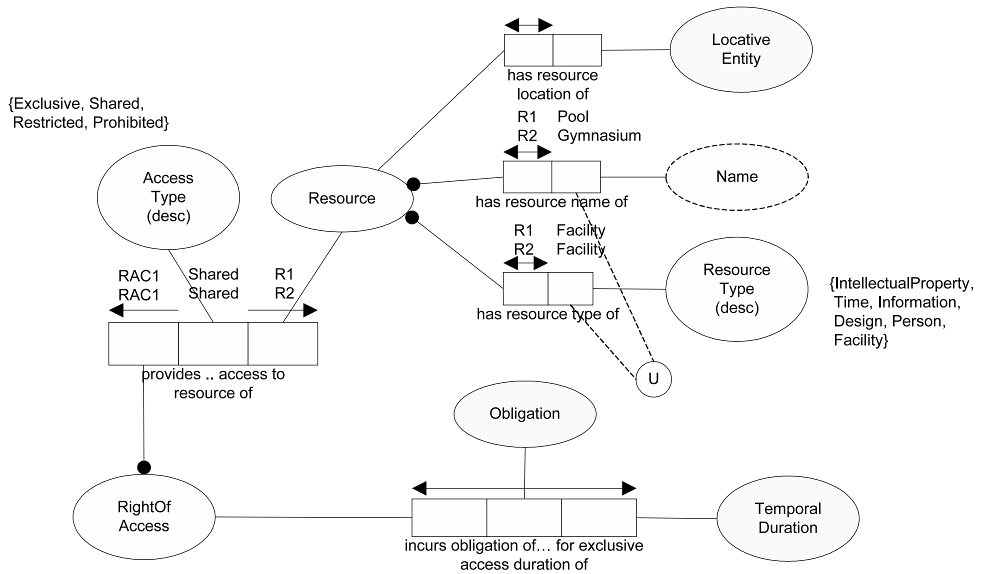
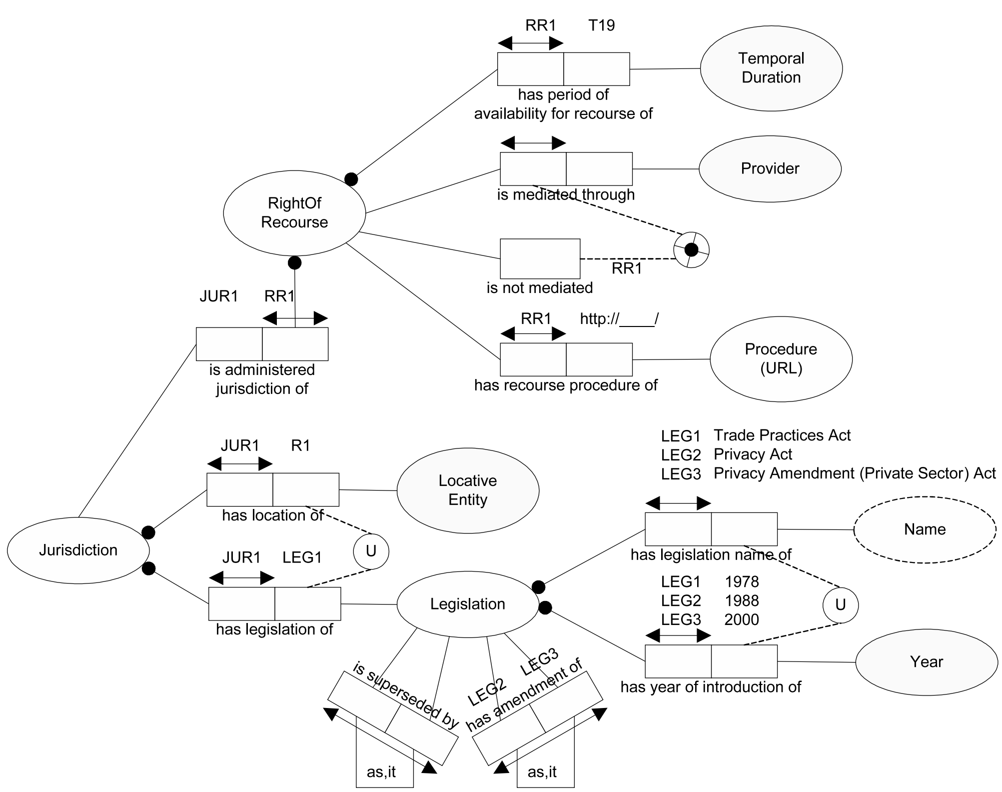


Figure 53: Rights.



each RightOfAccess **is a** Right **that** is of RightType 'X'

Figure 54: Right to access.



each RightOfRecourse **is a** Right **that** is of RightType 'C'

Figure 55: Right to recourse.

3.10.3 Suspension and resumption

We consider that two types of suspension exist [see Figure 55], voluntary and involuntary. Voluntary suspension is the ability of the service requestor to temporarily halt the provision of a service. Involuntary suspension results from the failure of a service requestor to meet obligations associated with a service. Involuntary suspension was outlined in section 3.9 relating to penalties. Both types of suspension may have an associated set of conditions, possibly incur a suspension or resumption charge, and have a procedure that outlines either or both the suspension or resumption process.

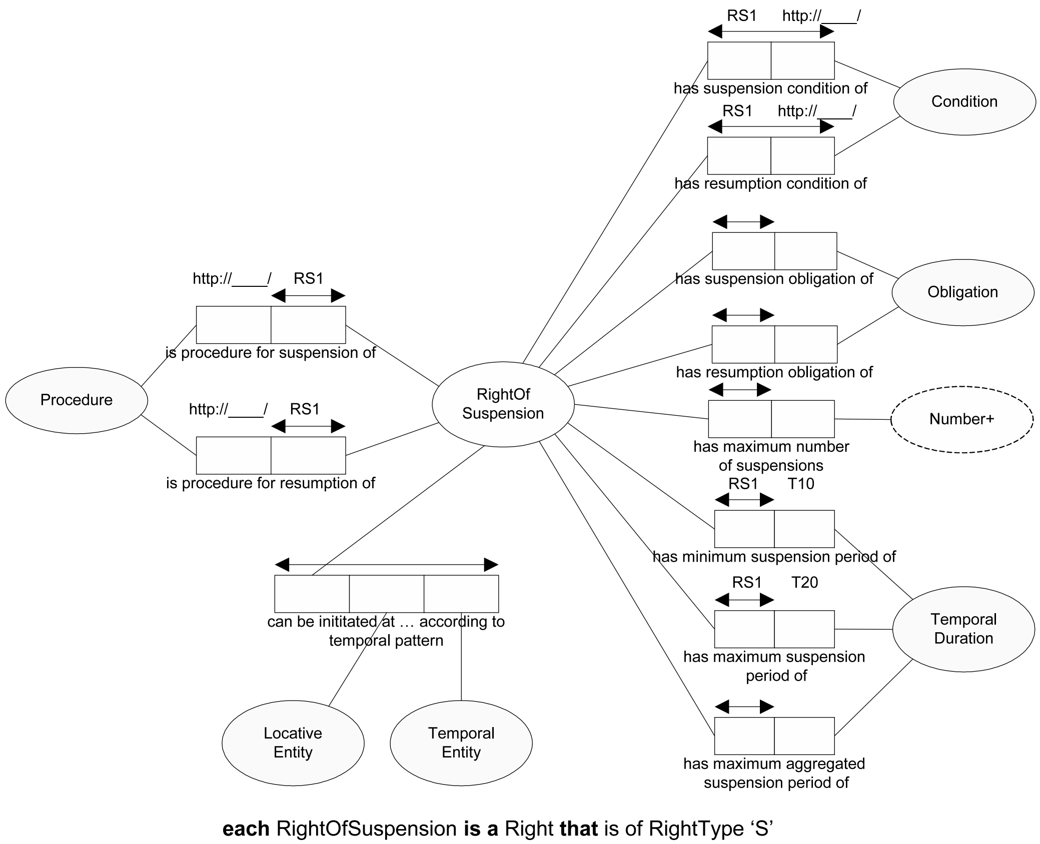


Figure 56: Right to suspend.

Within this section we are particularly interested in voluntary suspension. Voluntary suspension may be defined for specific periods of time. A service provider may set a minimum suspension period, a maximum suspension period, a maximum number of suspensions or even a maximum aggregated period of suspension that are available during the service provision process. The suspension of a service may normally be initiated at a certain locative entity, within the bounds of a temporal entity. Involuntary suspension results from the failure to meet one or more obligations. This type of suspension is invoked by the service provider under a set of conditions. The resulting suspension is normally for a specific suspension period.

3.10.4 Termination

Like suspension of a service, termination of the provision of a service can be initiated by either the service provider or the service requestor [see Figure 57]. Service providers will normally attempt to terminate a service commitment to a requestor when the requestor fails to comply with the obligations for a service. As with suspension we are particularly interested in termination from the voluntary perspective. Termination for failure to comply with an obligation is outlined in section 3.9 in relation to penalties.

Termination may incur an obligation (e.g. a cost), and may have a set of conditions surrounding its execution. The execution of the termination is normally governed by a termination procedure. The commitment to a service sometimes requires that termination be pre-conditioned by notification of the termination. This ensures that the party requesting the termination provides notification of the termination according to some temporal duration (e.g. 30 days notice).

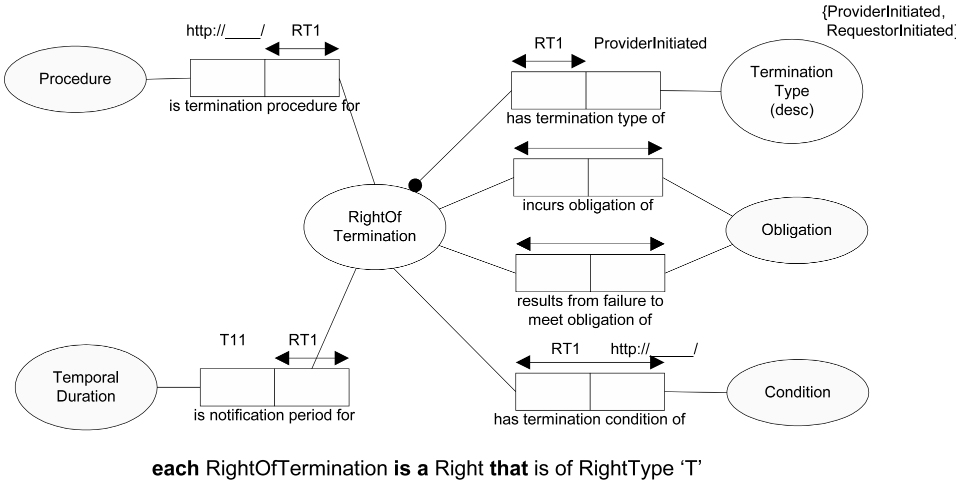


Figure 57: Right to terminate.

3.10.5 Privacy

Privacy is primarily the concern of the service requestor who wants to ensure that the service provider appropriately deals with information disclosed by the requestor to the provider [see Figure 58]. Privacy legislation sometimes binds the actions of a service provider when dealing with service requestor related information. These privacy concerns may be captured in legislation (e.g. a privacy act) or through a corporate privacy statement of the service provider.

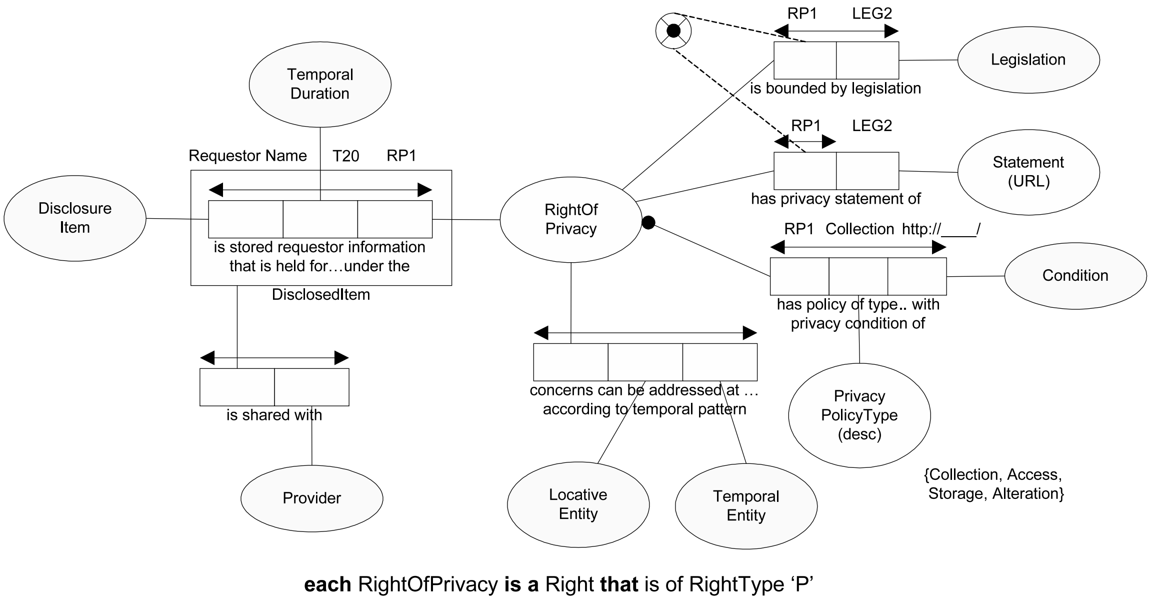


Figure 58: Right to privacy.

Privacy policies may be specifically defined for four distinct phases in the handling of service requestor information. These phases are the collection, storage, access and alteration of service requestor information. Concerns relating to a service provider's privacy commitment can normally be addressed at a specific location, within the bounds of a temporal entity. Service providers can outline other service providers who will be provided with certain requestor related information. These secondary service providers may be subsidiary companies of the original service provider. Privacy of disclosed items relating to the service requestor may be for a specific period. Subsequent to the expiry of that period the requestor must be consulted about the disclosure of the information.

3.10.6 Refusal of service

The right of refusal of provision is available to service providers [see Figure 59]. Service providers may outline the conditions under which service provision is refused, or may consider their right of refusal discretionary. The refusal of service provision may be enacted under a refusal procedure. Some service providers may offer an appeals procedure for requestors who are refused service. This procedure is normally available at a certain locative entity, within the bounds of a temporal entity.

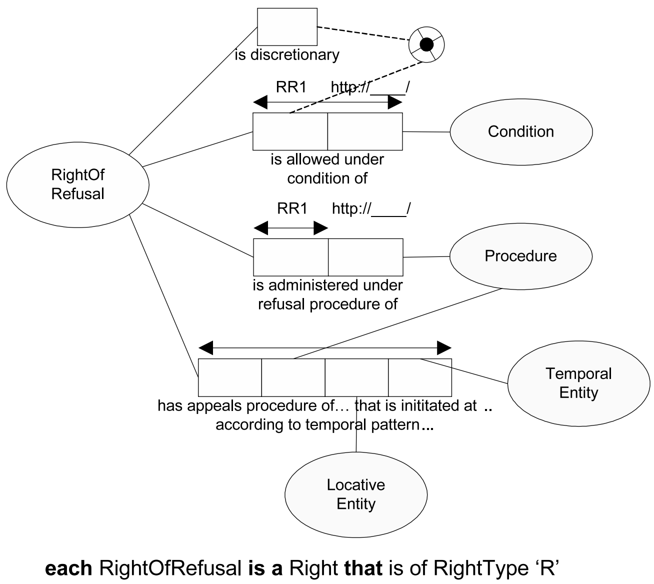


Figure 59: Right to refusal of service.

Refusal of access to a service is not to be confused with security of the service. When we refer to refusal we refer to notions such as nightclubs that retain the right to refuse entry to their services if patrons are inappropriately dressed, or who may have consumed excessive amounts of alcohol prior to their attempted entry.

3.10.7 Disclosure

Service providers occasionally request disclosure of information relating to the service requestor [see Figure 60]. This may be to allow them to create a database of target clients, or it may be to reduce their risk or exposure via a single client (as is the case with insurance providers). We refer to the information of interest to the service provider about the service requestor as disclosure items.

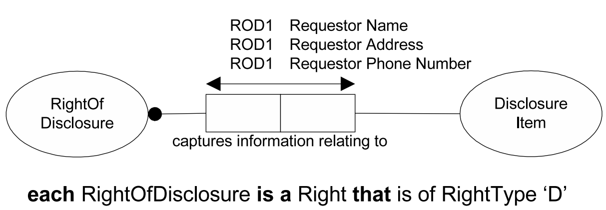


Figure 60: Right to disclosure.

Some service providers may expect disclosure of income, credit history, employment status etc. These facts about a service requestor are considered a disclosure item. The same entity "DisclosureItem" is used in a privacy context to outline the information about a requestor that is subject to privacy regulations/commitments. In this context it is primarily the service provider outlining that to use the service, requestors need to provide some information about themselves.

3.10.8 Extension

The right of extension to the service provision allows the service provider to advertise that provision may be extended for a certain duration under specific conditions [see Figure 61]. The extension to the provision may incur additional obligations (e.g. a lengthening of the original relationship commitment, or a cost). Extension of the provision is normally initiated via a defined procedure.

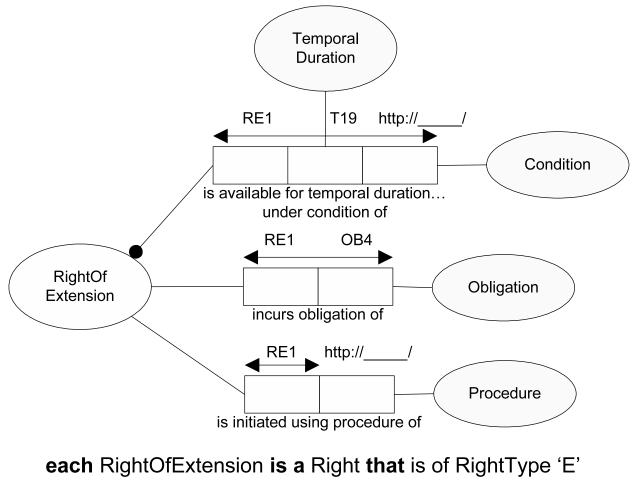


Figure 61: Right to extension of service provision.

Extension is common with leasing arrangements (e.g. for office space), and provides the service requestor with the option to exit their agreement should they not be satisfied.

3.10.9 Warranty

Warranties (or guarantees) enable a service provider to reduce the uncertainty surrounding the quality of service provision [see Figure 62]. Warranties are provided for specific item(s) and a specific period after the completion of service provision. They are surrounded by conditions, and may be fulfilled by a provider other than the original service provider.

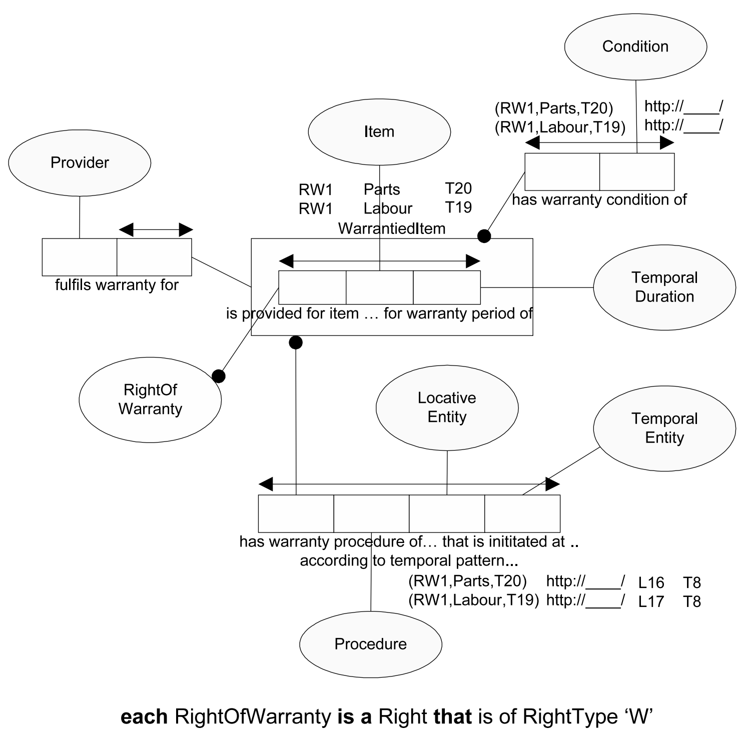


Figure 62: Right to warranty.

Warranties are normally initiated according to a procedure that is invoked at a specific location (i.e. a locative entity), and within some temporal parameters (e.g. a defined temporal interval). Warranties may be revoked when the obligations that a service requestor has with a particular service provider are not met (e.g. payment).

3.10.10 Cooling off period

Service requestors want a level of protection after requesting a service. This is referred to as a cooling off period [see Figure 63]. Normally, if a service offers a cooling off period they will be able to annul the service provision that they previously requested. By the term "annul", we are referring to the cancellation of all obligations that requestors have with the service provider. This annulment is normally available for a temporal duration (e.g. 7 days) after the request is made, or into the provision of the service. The latter is useful for services where the request and provision are non-contiguous. Our model also captures the conditions surrounding the cooling off period, and the procedure that needs to be invoked to ensure annulment of the service provision occurs.

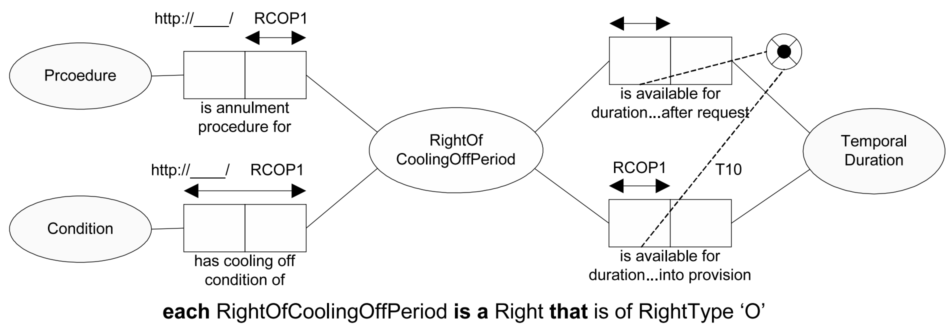


Figure 63: Right to cooling off period.

3.10.11 Liability limitation

Service providers want a level of protection from liability in the event of failure to provide a service as promised to the service requestor. We refer to this as limitation of liability [see Figure 64]. Our model provides an ability to capture the conditions under which the limits to liability can be expressed to the service requestor.

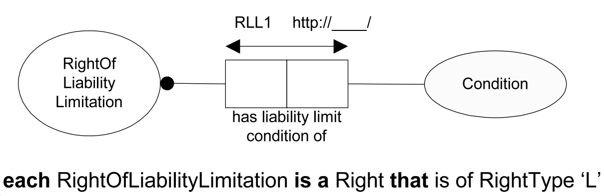


Figure 64: Right to limit liability.

3.10.12 Registered intellectual property

Rights may take the form of registered intellectual property (IP) rights that a service provider holds. This includes common intellectual property rights such as patents, trademarks and designs. We discuss registered intellectual property rights generically before dealing with the specific subtypes of trademarks, patents and designs. All registered property rights are owned by a service provider. They will commonly have the following related details [see Figure 65]:

- ◇ An application number (assigned when it is lodged).
- ◇ Status - A status for the application of the intellectual property right. Applications for intellectual property rights move through various statuses. These indicatively include applied, published, examined, approved and rejected.
- ◇ Lodgement date - The date on which the application for registration of the intellectual property right was lodged.
- ◇ Registration date - The date on which the application was granted for the intellectual property right.
- ◇ Description - A link to a more detailed description of the intellectual property right. Normally this would be a URI pointing to a patent/trademark agency such as the United States Patent and Trademark Office.
- ◇ Country of origin - The country within which the intellectual property right originated.

◇ Region of protection - The area within which a granted intellectual property right applies.

◇ Agent - The representative (normally legal) for the party making the application.

◇ Agent's address - The address of the agent for the party making the application.

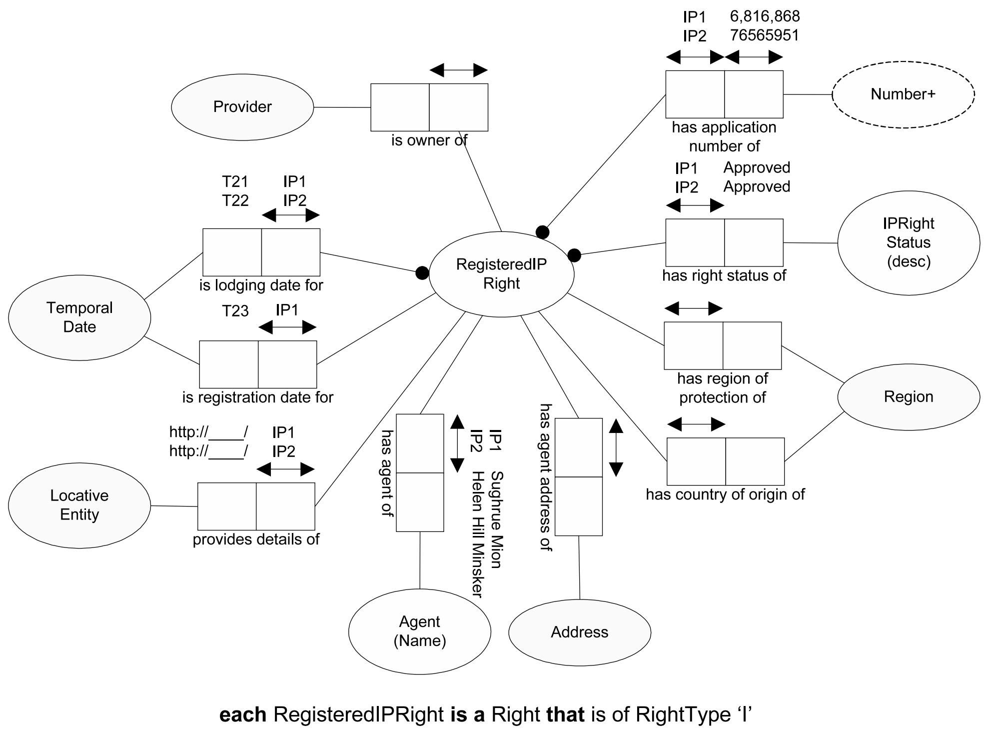


Figure 65: Registered IP rights.

Instances of a registered intellectual property right are one of the following types: trademark, patent or design [see Figure 66]. We also include a TrademarkOrDesign entity that captures the duration of protection granted to trademarks and designs. This property is inherited by both trademarks and designs.

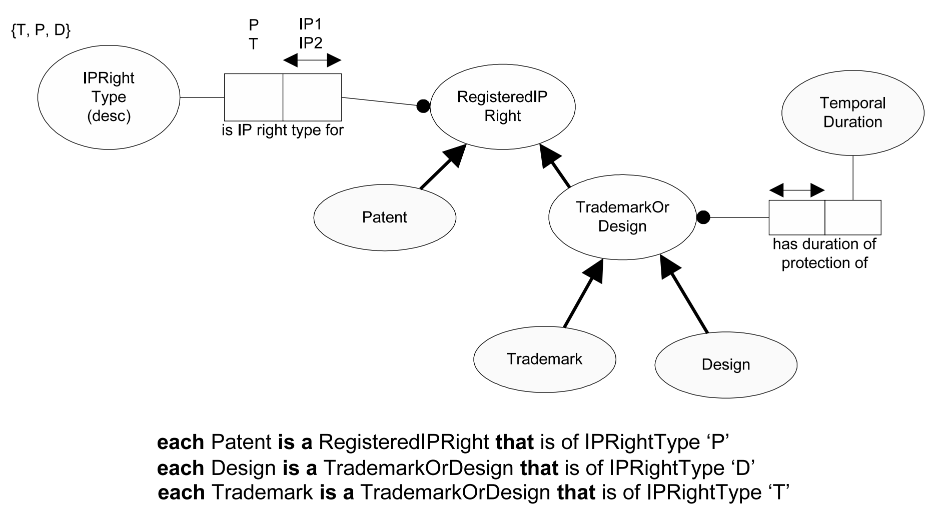


Figure 66: Registered IP right subtypes.

Patents are official rights granted to an inventor with respect to an invention [25]. Within our patent model we capture the name of the inventor of the patent, and the title of the patent. We provide support for the classification of patents according to the Strasbourg Agreement Concerning International Patent Classification [35]. This agreement, signed by member countries, provides for a classification of patents according to a hierarchical identifier that includes a section (identified using letters A - H), a class (identified using a two digit number), a

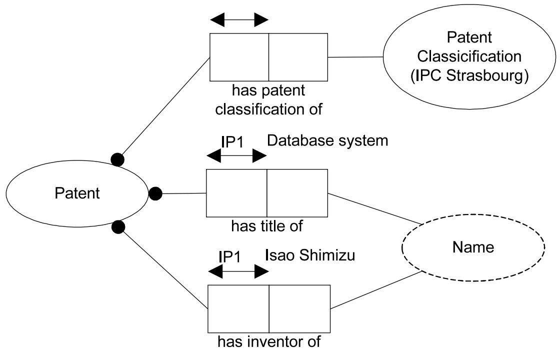


Figure 67: Patents.

subclass (identified by a capital letter), and a group (identified using two, two digit numbers separated with a forward slash).

Trademarks are "a word, name, symbol or device that is used in trade with goods to indicate the source of the goods and to distinguish them from the goods of others" [25]. Trademarks are further divided into the categories of trademark, servicemark, dressmark, collectivemark or certification mark. Servicemarks are similar to trademarks except that they apply to a service, not a product. Trademarks have a wordmark that is outlined at a URI location.

Classification of trademarks is supported through the figurative marks contained within them [34], and the goods or services that the trademark applies to [33]. The former classification scheme uses a category number (1 - 29) and division (identified by a two digit number). The latter approach involves 45 categories, the first 34 of which are related to goods. The remaining categories relate to services.

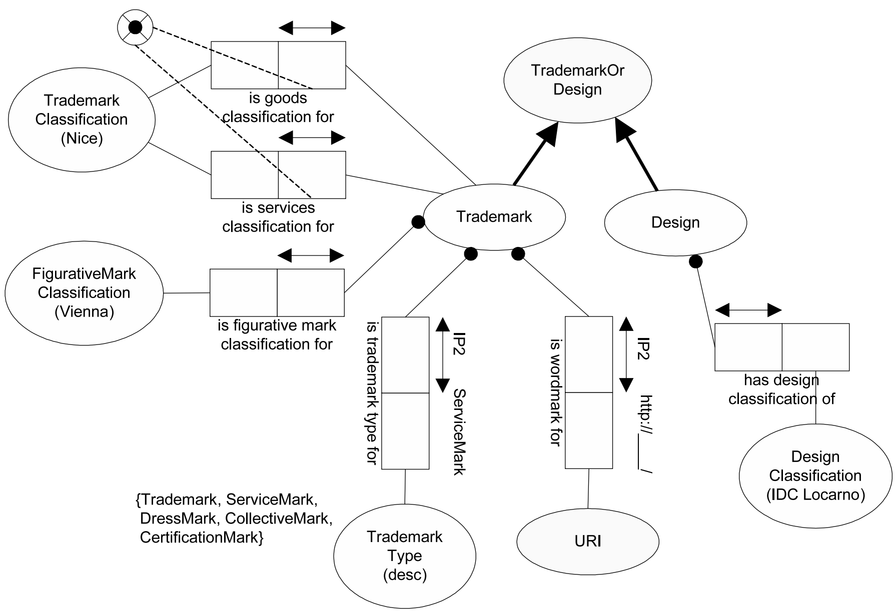


Figure 68: Trademarks and designs.

A design consists of the "overall appearance of a product", that when considered with respect to its "shape, configuration, pattern, and ornamentation" give it a "unique appearance" [4]. Design may also be classified according to the International Design Classification [36]. The scheme, referred to as the Locarno classification, provides a 32 class and 223 sub-class categorisation.

3.11 Language

We offer three types of support for language within our model [see Figure 69]. We attach this support for language to the locative entity which the service requestor interacts with. The three types of language support are:

- ◇ That a service is capable of interacting with a service requestor in a written manner using a language defined within the ISO639-2 standard [27]. ISO639-2 uses a three letter character code to represent a language, or a family of languages.

- That a service is capable of interacting with a service requestor in a verbal manner using a language defined within ISO639-2; or
- Finally, that a service is capable of interacting with a service requestor according to a standard (e.g. Web Services Description Language - WSDL).

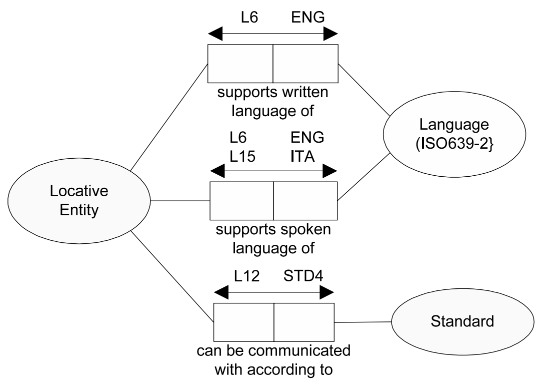


Figure 69: Language.

Examples of ISO639-2 codes include: “afr” for Afrikaans, “eng” for English, “fin” for Finish and “ita” for Italian. In some instances, ISO639-2 uses a combination of two three-letter codes to describe the language. The first refers to a bibliographic code (sometimes termed ISO639-2/B) and the latter refers to a terminology code (sometimes referred to as ISO639-2/T). We choose not to distinguish between the type of code (i.e. bibliographic or terminology) and provide the capability to store multiple ISO639-2 codes against a single locative entity.

Our final type of support, that treats locations as being capable of communicating according to a standard, whilst appearing cursory caters for the storage of multiple standards against a single location. This is useful when two or more types of standards are necessary for description of the service. For example, WSDL describes the operations and messages that a web service exposes. We could additionally attach a standard that describes the choreography of messages over-and-above the WSDL.

3.12 Trust

Our view is that trust is largely indirectly represented in the models that are outlined in this paper. We believe that people’s understanding and requirements for trust vary. Accordingly, some of the items that we feel contribute to a person’s level of trust when dealing with both the service provider (in general) and the service include:

- The cost of the service in comparison to the charge being levied by another service provider.
- The security of the payment instrument to be used (e.g. some credit cards offer a chargeback facility).
- The security of the location for payment.
- Endorsement received from previous service requestors, or other service providers who compose services using the service in question.
- The support that a service has for certain quality standards.
- The membership of the service provider to a certain body (perhaps a professional body).
- The number of years that the service provider has been in business, or the number of years that the service has been offered.
- The mission statement of the service provider.
- The laws that govern the business operating environment.
- The links that this provider has with other providers who are perhaps better known to service requestors.

◇ The privacy of the information supplied to the service provider.

◇ The openness of the service provider to the description process.

We choose to directly represent the notion of endorsement as a means of providing "referrals" from service requestors (or perhaps service providers) to the service in question [see Figure 70]. We consider endorsements to include the capture of the service party (presented in Figure 1) providing the endorsement, some comments relating to the service party's use of the service, a temporal instant when the endorsement was captured (providing context), and a subjective rating of the service provider/service by the party giving the endorsement. Ratings are discussed in more depth in section 3.13.1.

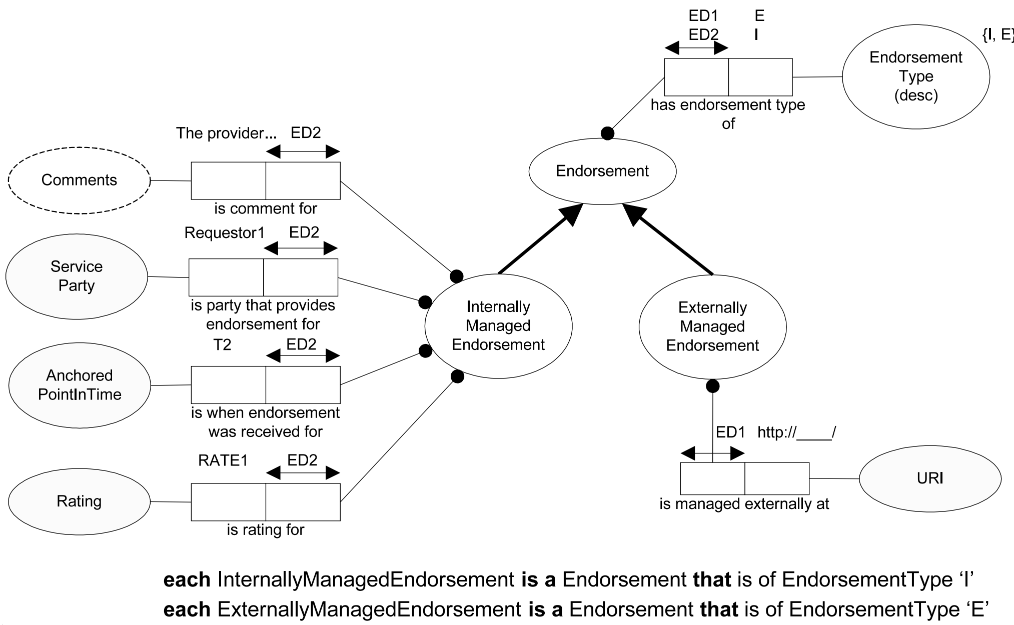


Figure 70: Endorsement.

We view endorsements as being either managed internally by the provider offering the service, or being managed externally. The external management may imply a less biased view of the endorsements for a service. A provider who internally manages endorsements may be inclined to only show those that discuss their service in a favourable manner. For those services where their endorsements are managed externally, we provide a link to the location where those endorsements are managed. It is likely that this would be used to refer to a URI.

We directly represent endorsements in our model by linking it to both the service provider and the service. As well as supporting the notion of endorsements for service providers [see Figure 71], we also capture the legislation that a provider is legally bound to comply with, the year that the service provider began business, the mission statement of the provider, the type of associations that it has with other providers, and the memberships that it holds with certain bodies. We capture associations such as the service provider being a partner of another provider, being a subsidiary, an owner, a supplier to, an agency, a division and a branch.

The non-functional properties of trust that we directly represent with a service include two previously mentioned (the year of inception and links to endorsements received), and whether the payment obligation for a service can be executed using a particular escrow or insurance service [see Figure 72].

The cumulative result of both our direct and indirect models with respect to trust is a domain independent approach that allows service requestors to determine the non-functional properties of services that best fulfil their requirements for trust. The following example query filters instances of directly modelled trust for a service provider and a service based on the following criteria: the service inception is prior to 1990, the service provider inception is prior to 1985, and the provider is a partner with "Microsoft".

Provider
  ⊢ has year of inception 1985
  ⊢ has association of type "Partner" with Provider
    ⊢ has provider name "Microsoft"
  ⊢ offers Service
    ⊢ has year of inception < 1990

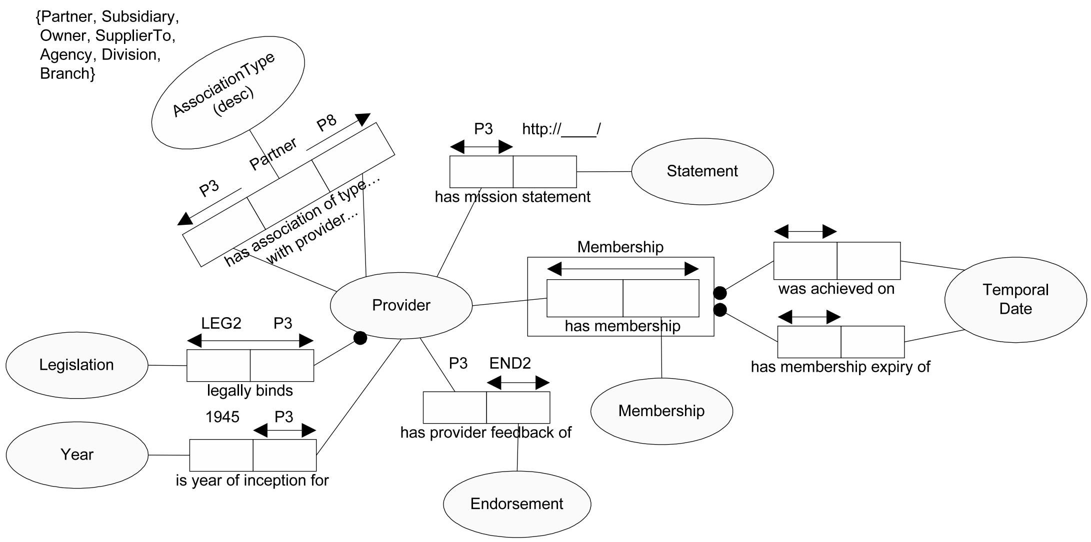


Figure 71: Provider related trust.

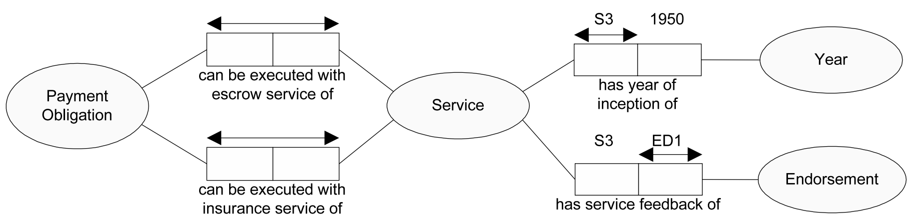


Figure 72: Services related trust.

3.13 Quality

Representing quality of service is difficult whilst trying to maintain a domain-independent view of service description. We take the view that services might prefer to capture quality with respect to a standard, an industry benchmark and/or a ranking scheme. The latter two approaches also allow a provider to state a comparative assessment of their service with respect to an industry benchmark, whether it be a self-assessment or an independently verified assessment of their conformance to the standard. We differentiate between industry benchmarks and rating schemes in the following manner. We consider industry benchmarks to be similar to a survey of service providers. These surveys capture certain indicators about the service. For example, a hotel industry benchmark may capture an indicator such as revenue per available room or average room rate per night. Ranking schemes we consider are a form of ranking of a service. For example, restaurants can be ranked in the Michelin Guide [19] according to the number of Michelin stars that they have received (between 1 and 3).

We have used existing work [37] as the basis for determining the aspects of service quality. They outline eleven (11) dimensions of perceived electronic service quality. We subscribe to Zeithaml's notions of reliability, responsiveness, access, efficiency, assurance/trust, security/privacy, price knowledge, customisation and believe that they can be captured either using the approach outlined above, or are dealt with in detail in another model. We consider the dimensions of ease of navigation and site aesthetics to be specific to electronic services. As we are trying to capture both electronic and traditional services we have ignored these dimensions.

Expectation of quality is importantly distinguished in [17] as occurring before the service interaction. We provide support for the aforementioned dimensions by including their description within our models. The expectation of service quality is therefore based on the description that is provided to the service requestor by the service catalogue. Whilst all the dimensions encourage expectations, the final dimension of flexibility appears to be dependent on some of the non-functional properties such as the ways to pay, buy or return items. It also includes notions such as the way to search for items. We believe that a service that offers sufficient flexibility to a service requestor in terms of the options available to them will determine a requestor's view of quality with respect to flexibility.

3.13.1 Standard, benchmark and ranking schemes

We use the term standard to refer to items "established or widely recognized as a model of authority or excellence" [30] [see Figure 73]. Standards are commonly published by service industries (captured in our model using a UN/SPSC code [21]) or a service provider. They are regularly referred to using a title name and a publication date. Standards also have one or more author names, a version number, and a status (reflecting whether they are in a draft or final stage of completion). We provide a link to a location where the standard may be accessed, a reference to a standard superseding another standard, and that a standard offers differing levels of conformance. This allows us later to refer to the comparison of the service provider to the criterion within the standard using the standard level. We provide the means of nominating one or more regions as being covered by the standard.

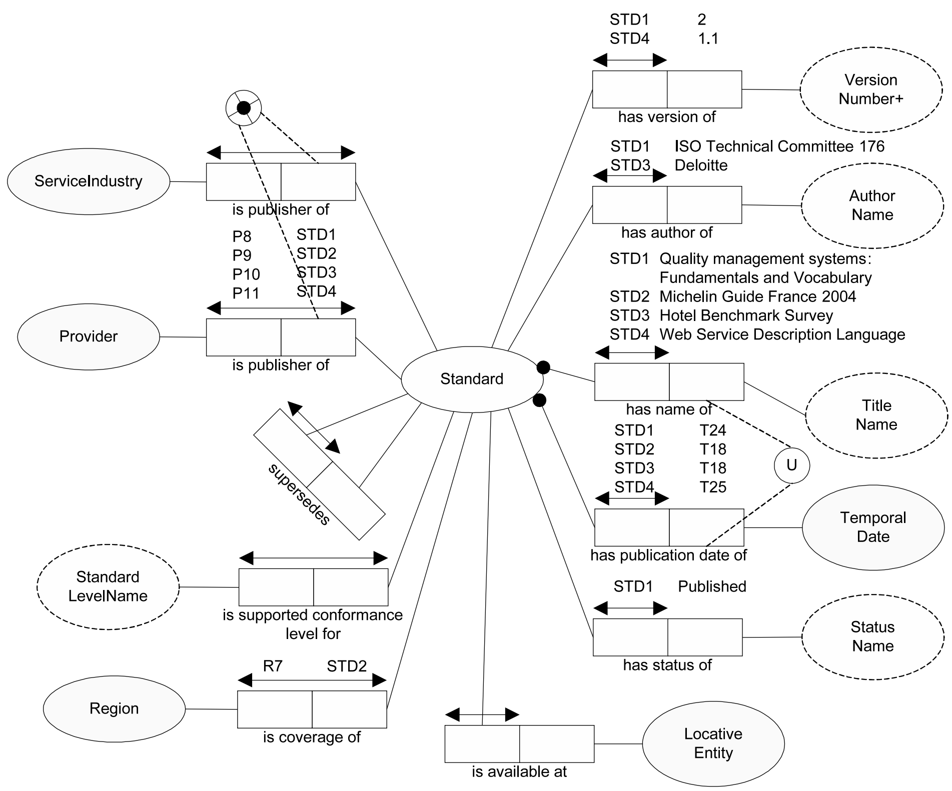


Figure 73: Standards.

We further subtype standards into assessment standards. As previously mentioned this allows us to state a comparative assessment with respect to a standard for a service provider. Industry benchmarks are analogous to a survey, whilst we consider rating schemes to be a form of ranking for a service [see Figure 74].

We consider benchmarks to have one or more indicators. An indicator can be thought of as a key performance indicator (KPI) or similar, that tracks a particular item (e.g. average room rate per night). Indicators may apply during a certain temporal period or duration, may be valid for a particular region (e.g. a state within a country), and percentage change from the previous indicator result. Ranking schemes have one or more rankings that are normally valid for a specific period (e.g. 12 months) and have a ranking position (e.g. we refer to this using an integer value).

We have not expressed the specific information about the indicator or the ranking. We have subtyped these two entities from a super-type that we refer to as a rating [see Figure 75]. We consider ratings to have a name (the item that we are referring to) and a description. Ratings have a rating value that is an expression of the rating as either a percentage, a numeric value, a price, or a string representation.

Having shown how we capture the information relating to standards we can now express a comparative assessment with respect to the standard for the service [see Figure 76]. We provide for assessments for three quality dimensions: reliability, responsiveness and efficiency of the service. The service can capture the assessment of a particular dimension using a rating value of a particular rating with an assessment standard. Additionally, we capture that this comparative assessment was independently verified by another provider, as well as the date it was achieved.

In addition to a comparative assessment, we consider that quality of a service may require some form of feedback endorsement for a particular dimension of the service, that it allows configuration of a user profile at

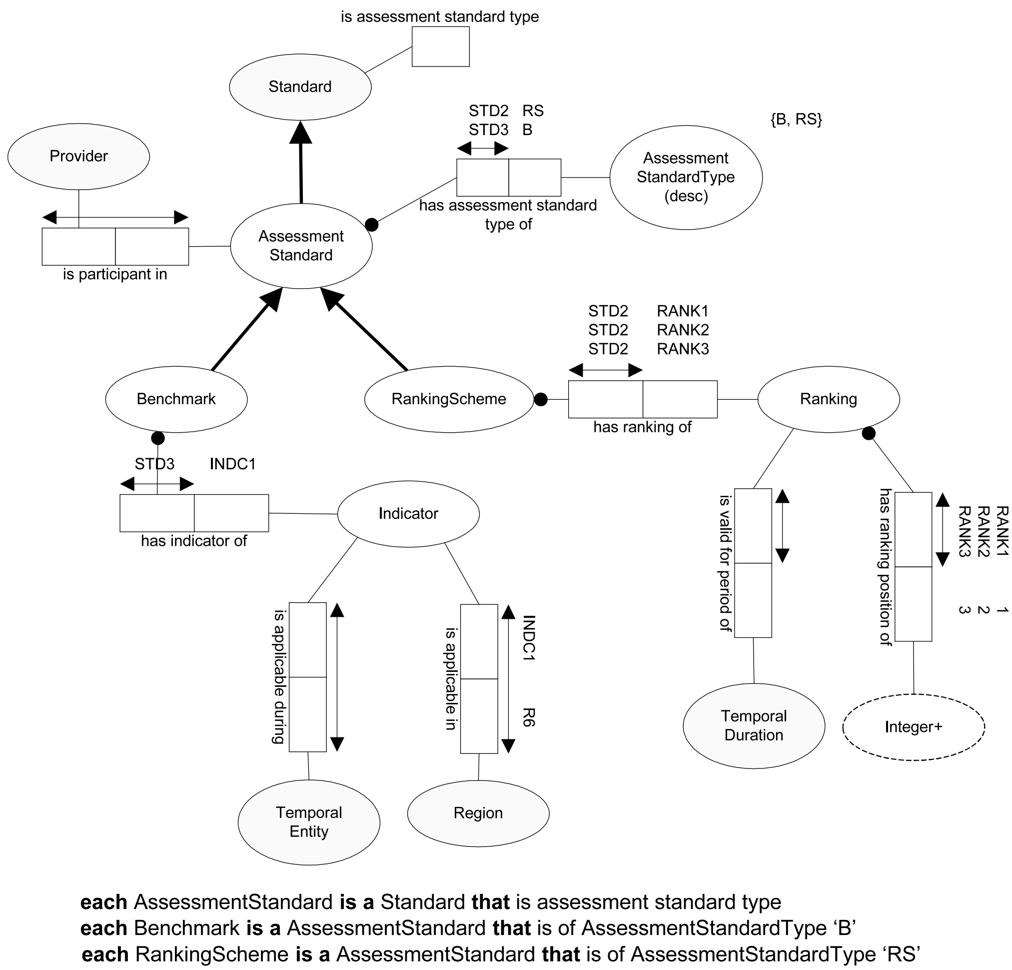


Figure 74: Benchmarks.

a certain location, and that it captures an interaction history. Providers may also achieve conformance of a standard for a service on a particular date. This compliance has a conformance rating.

3.14 Security

We attach the non-functional property of security to the locative aspects of a service (i.e. where it can be requested, provided, paid for, queried for further information etc). This is due to the assumption that interactions that occur between the service provider and the service requestor must occur at a location or over some form of communications medium). We include the communications medium within this definition as there may be a distance between the service provider and the service requestor (e.g. a web service request). With neither of these dimensions do we attempt to describe the security of the service in-depth. We prefer to provide an overview of the type of security that the service requestor can expect.

We divide our discussion of security into two dimensions: identification and confidentiality. Examples of identification requirements for services include:

- ◇ Automated Teller Machine (ATM) Access to an ATM normally occurs using a card and a personal identification number (PIN). This can be considered the "something you have" and the "something you know" principles that together provide sufficient information for the ATM to determine if you have the ability to access the ATM.

- ◇ Web Site Access Some web sites are secured with a username (or customer number) and password combination, possibly just a password. Examples include online banking, newspaper web sites, and web-based email.

- ◇ Biometric - Some services require biometric related information to determine if a requestor has access to a service. Biometric information is categorised as appearance, social, natural, biodynamics and imposed characteristics [9]. Common biometric security uses finger print, retinal scan and facial scan technologies.

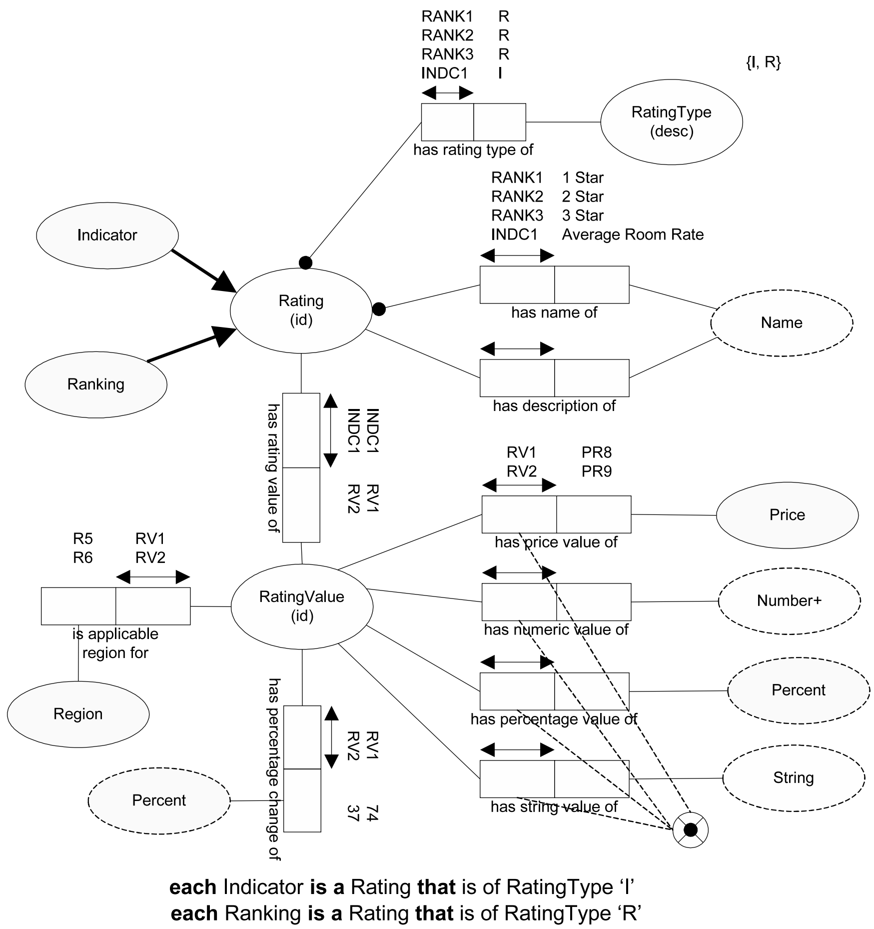


Figure 75: Ratings.

We present our model for security identification in Figure 78. We stipulate that a location may have one or more identification requirements. The identification requirement will be composed of a mandatory set of identification types that can be presented by the service requestor, as well as additional identification item. It is possible for the service provider to allocate a number of identification points per identification requirement /identification type combination (referred to in the model as AcceptableIdentification), as well as stipulating that the identification requirement must meet a collective points total to be achieved.

We consider identification types to include (but not be limited to): PIN, password, username, birth certificate, marriage certificate, membership card, licence, passport, student card, physical key, X509 certificate, Kerberos ticket, debit card, credit card, or a national identity card. Personal information such as names, addresses, dates of birth, or phone numbers are sometimes used to validate claims for access. In a WS-Security context this is referred to as proof of possession, and is used to validate unendorsed claims [20]. An endorsed claim is backed by the presentation of an X509 certificate or a Kerberos ticket. These identification types have been endorsed by another party (e.g. Verisign). Other identification types include biometric based mechanisms: facial scan, retinal scan, iris scan, thumprint, voice, finger print set, hand geometry and DNA. A table similar to the various characteristics of payment instruments could be constructed for these identification types.

The second dimension of security that we model is confidentiality [see Figure 79. Confidentiality has two connotations depending on the domain. Firstly, it can be considered as the authorising of access or as an ethical principle that must be adhered to (e.g. in fields like medicine or journalism). We consider that this type of confidentiality is normally formalised into a confidentiality agreement. These types of agreements normally have a temporal interval that defines when they are applicable, and a legal jurisdiction within which it is controlled. Finally, we consider that confidentiality is applicable to the communications mechanism over which a service provider and service requestor interactions occurs. We capture the name, the key length and provide a link to the standard that defines the details of the encryption technique (or algorithm).

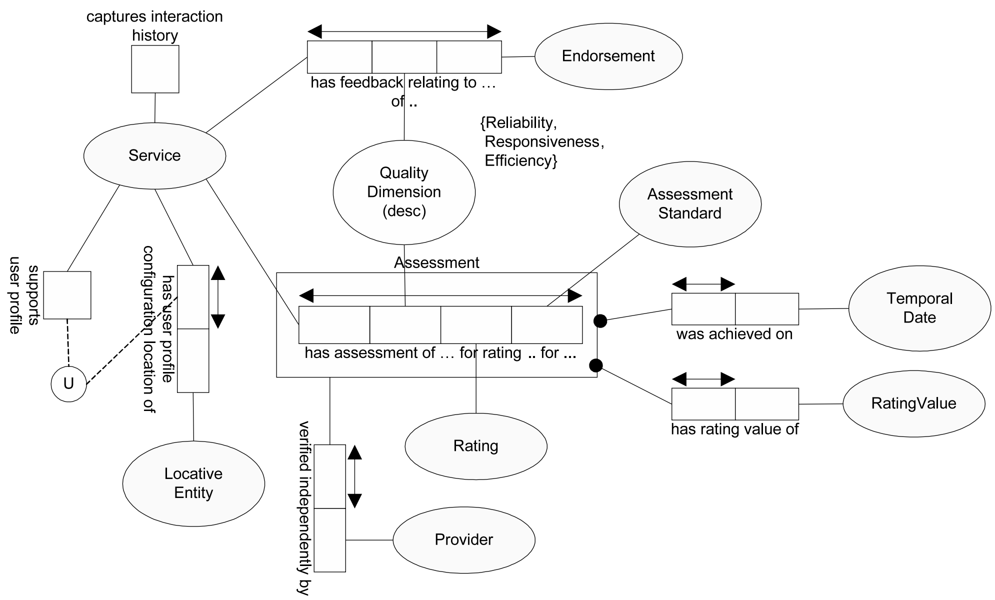


Figure 76: Service quality.

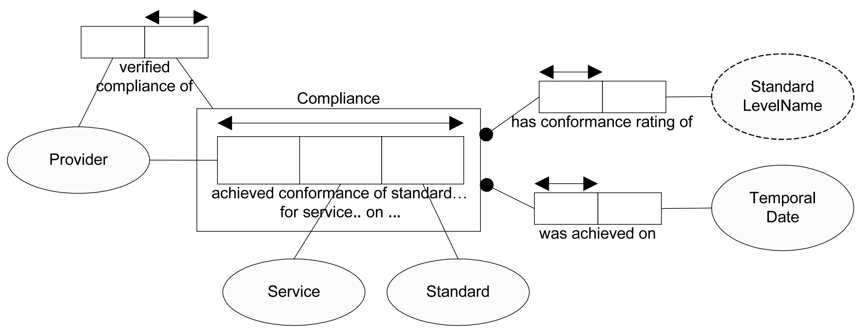


Figure 77: Provider quality.

4 Model example

This section provides a detailed outline of the service example that was presented in section 1.2.1. We present it in the form of a verbalisation of the ORM schemas that apply to the example.

- Service provider verbalisations:
  - 'S1' (Service) provides capability of (Capability) 'C1'.
  - 'S1' is referred to by 'Swan Lake'.
  - 'S1' is offered by (Provider) 'P1'.
  - 'P1' has provider name 'Unnamed Ballet Company'.
  - 'P' (ServicePartyType) is service party type of 'P1'.
  - 'P1' operates in industry '90150000'.
  - 'P2' has provider name 'American Express'. (see payment obligation verbalisations).
  - 'P' (ServicePartyType) is service party type of 'P2'.
  - 'P3' has provider name 'Diners'. (see payment obligation verbalisations).
  - 'P' (ServicePartyType) is service party type of 'P3'.
- Temporal verbalisations are defined here for later use with respect to availability:
  - 'A' is temporal entity type for (AnchoredPointInTime) 'T1'.
  - 'T1' has hours of '19'.
  - 'T1' has minutes of '30'.

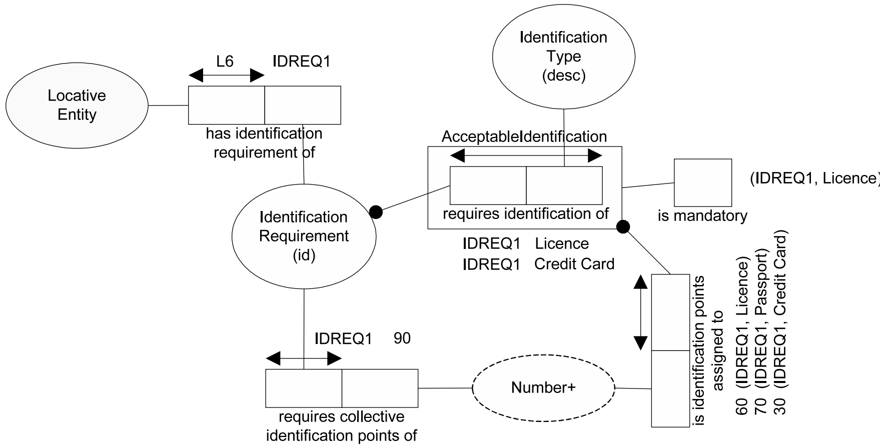


Figure 78: Identification.

– ‘T1’ has seconds of ‘0’.
– ‘T1’ has timezone of (Timezone) ‘TZ1’.
– ‘TZ1’ has region of ‘R1’. (see locative verbalisations).
– ‘TZ1’ has UTC offset of ‘+10:30’.
– ‘CalDate’ is temporal entity type for (CalendarDate) ‘T2’.
– ‘T2’ has year ‘2005’.
– ‘T2’ has month number of ‘2’.
– ‘T2’ has day of month number of ‘24’.
– ‘T1’ has calendar date of ‘T2’.
– ‘A’ is temporal entity type for (AnchoredPointInTime) ‘T3’.
– ‘T3’ has hours of ‘19’.
– ‘T3’ has minutes of ‘30’.
– ‘T3’ has seconds of ‘0’.
– ‘T3’ has timezone of (Timezone) ‘TZ1’.
– ‘CalDate’ is temporal entity type for (CalendarDate) ‘T4’.
– ‘T4’ has year ‘2005’.
– ‘T4’ has month number of ‘2’.
– ‘T4’ has day of month number of ‘25’.
– ‘T3’ has calendar date of ‘T4’.
– ‘A’ is temporal entity type for (AnchoredPointInTime) ‘T5’.
– ‘T5’ has hours of ‘13’.
– ‘T5’ has minutes of ‘30’.
– ‘T5’ has seconds of ‘0’.
– ‘T5’ has timezone of (Timezone) ‘TZ1’.
– ‘CalDate’ is temporal entity type for (CalendarDate) ‘T6’.
– ‘T6’ has year ‘2005’.
– ‘T6’ has month number of ‘2’.
– ‘T6’ has day of month number of ‘26’.
– ‘T5’ has calendar date of ‘T6’.
– ‘A’ is temporal entity type for (AnchoredPointInTime) ‘T7’.
– ‘T7’ has hours of ‘19’.
– ‘T7’ has minutes of ‘30’.

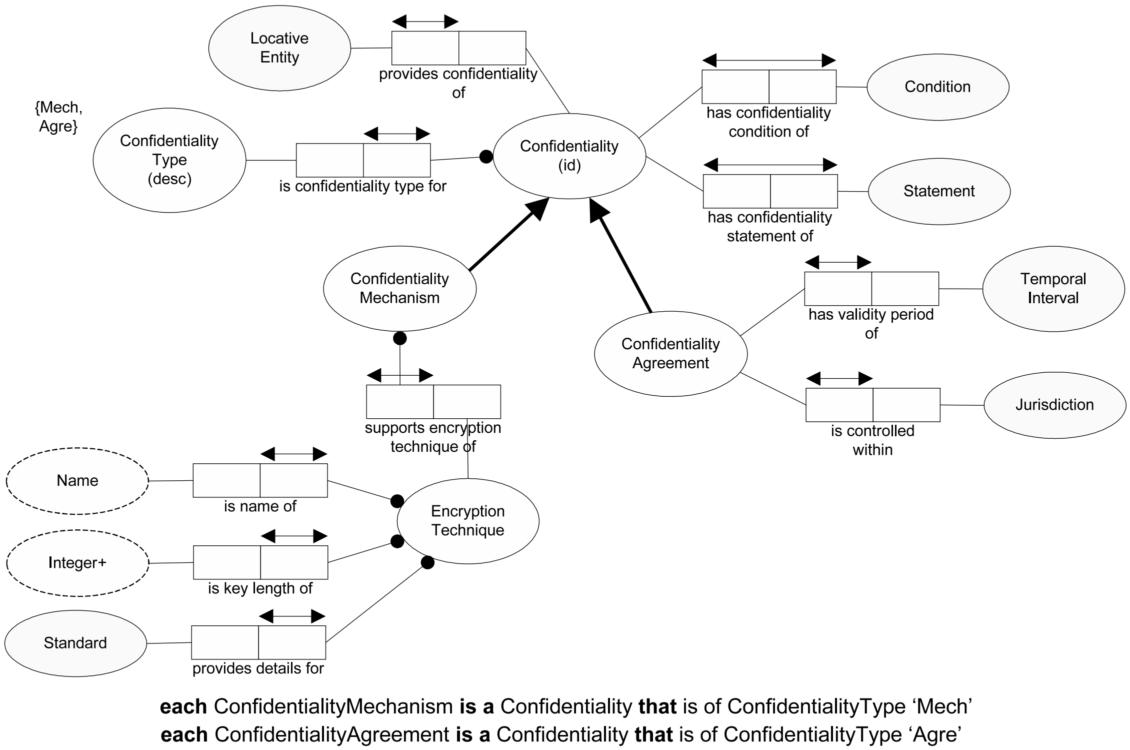


Figure 79: Confidentiality.

- 'T7' has seconds of '0'.
- 'T7' has timezone of (Timezone) 'TZ1'.
- 'T5' has calendar date of 'T6'.
- 'A' is temporal entity type for (AnchoredPointInTime) 'T8'.
- 'T8' has hours of '18'.
- 'T8' has minutes of '30'.
- 'T8' has seconds of '0'.
- 'T8' has timezone of (Timezone) 'TZ1'.
- 'CalDate' is temporal entity type for (CalendarDate) 'T9'.
- 'T9' has year '2005'.
- 'T9' has month number of '2'.
- 'T9' has day of month number of '28'.
- 'T8' has calendar date of 'T9'.
- 'A' is temporal entity type for (AnchoredPointInTime) 'T10'.
- 'T10' has hours of '19'.
- 'T10' has minutes of '30'.
- 'T10' has seconds of '0'.
- 'T10' has timezone of (Timezone) 'TZ1'.
- 'CalDate' is temporal entity type for (CalendarDate) 'T11'.
- 'T11' has year '2005'.
- 'T11' has month number of '3'.
- 'T11' has day of month number of '1'.
- 'T10' has calendar date of 'T11'.
- 'A' is temporal entity type for (AnchoredPointInTime) 'T12'.
- 'T12' has hours of '19'.

  - ‘T12’ has minutes of ‘30’.
  - ‘T12’ has seconds of ‘0’.
  - ‘T12’ has timezone of (Timezone) ‘TZ1’.
  - ‘CalDate’ is temporal entity type for (CalendarDate) ‘T13’.
  - ‘T13’ has year ‘2005’.
  - ‘T13’ has month number of ‘3’.
  - ‘T13’ has day of month number of ‘2’.
  - ‘T12’ has calendar date of ‘T13’.

- Locative verbalisations are defined here for later use with respect to availability:
  - ‘R1’ (Region) has locative entity type of ‘R’.
  - ‘R1’ has region type of ‘Country’.
  - ‘R1’ has region name of ‘Australia’.
  - ‘A1’ (Address) has locative entity type of ‘SA’.
  - ‘A1’ has common name of ‘QPAC’.
  - ‘A1’ has common name of ‘Lyric Theatre’.
  - (‘A1’, ‘QPAC’) (LocativeCommonName) applies to location ‘R2’.
  - (‘A1’, ‘Lyric Theatre’) (LocativeCommonName) applies to location ‘R2’.
  - ‘R2’ (Region) has locative entity type of ‘R’.
  - ‘R2’ has region type of ‘City’.
  - ‘R2’ has region name of ‘Brisbane’.
  - ‘A1’ has country of ‘R1’.
  - ‘A1’ has city of ‘R2’.
  - ‘A1’ has postcode of ‘4101’.
  - ‘A1’ has street name of ‘Melbourne’.
  - ‘A1’ has street type of ‘Street’.
  - ‘PH1’ (PhoneNumber) has locative entity type of ‘M’.
  - ‘PH1’ has phone number type of (PhoneNumberType) ‘FixedLine’.
  - ‘PH1’ supports interaction type of (InteractionType) ‘Voice’.
  - ‘PH1’ has local number of ‘136246’.
  - ‘URI1’ (URI) has locative entity type of ‘U’.
  - ‘URI1’ has scheme type of (SchemeType) ‘http’.
  - ‘URI1’ has authority of (Authority) ‘AUTH1’.
  - ‘AUTH1’ has server name of ‘www.qtix.com.au’
  - ‘URI1’ has path of ‘/’.
  - We have not outlined all Qtix outlets here. Each address would be defined in a manner similar to the ’A1’ street address entity.
  - ‘URI2’ (URI) has locative entity type of ‘U’.
  - ‘URI2’ has scheme type of (SchemeType) ‘http’.
  - ‘URI2’ has authority of (Authority) ‘AUTH2’.
  - ‘AUTH2’ has server name of ‘www.unnamedballetcompany.com.au’
  - ‘URI2’ has path of ‘/’.

- Service provision availability verbalisations:
  - ‘PL1’ (ProvisionLocation) is offered for service ’S1’.
  - ‘A1’ is location that service can be provided to ’PL1’.
  - (‘PL1, S1’) (ProvisionLocativeAvailability) can be provided during ‘T1’.

- (‘PL1, S1’) can be provided during ‘T3’.
- (‘PL1, S1’) can be provided during ‘T5’.
- (‘PL1, S1’) can be provided during ‘T7’.
- (‘PL1, S1’) can be provided during ‘T8’.
- (‘PL1, S1’) can be provided during ‘T10’.
- (‘PL1, S1’) can be provided during ‘T12’.

◇ Service request availability verbalisations:

- ‘S1’ accepts request of type ’Capability’ (RequestType) at ‘A1’.
- ‘S1’ accepts request of type ’Capability’ at ‘PH1’.
- ‘S1’ accepts request of type ’Capability’ at ‘URI1’.
- (‘S1’, ‘Capability’, ‘PH1’) (RequestLocativeAvailability) is continuously available for requests.
- (‘S1’, ‘Capability’, ‘URI1’) is continuously available for requests.
- ‘S1’ accepts request of type ’Information’ at ‘URI2’.

◇ Obligation related verbalisations:

- ‘S1’ has obligation of ‘OBG1’ for service provider ‘P1’.
- ‘S1’ has obligation of ‘OBG2’ for all service provider requestors.

◇ Pricing obligation verbalisations:

- ‘OBG1’ (PricingObligation) has obligation type of ‘C’.
- PricingObligation ‘OBG1’ results in price of (Price) ‘PRICE1’.
- ‘PRICE1’ (AbsolutePrice) has price type of (PriceType) ‘Abs’.
- ‘PRICE1’ has amount of ’85’.
- ‘PRICE1’ has currency of ’AUD’.
- ‘PRICE2’ (AbsolutePrice) has price type of (PriceType) ‘Abs’.
- ‘PRICE2’ has amount of ’75’.
- ‘PRICE2’ has currency of ’AUD’.
- ‘PRICE3’ (AbsolutePrice) has price type of (PriceType) ‘Abs’.
- ‘PRICE3’ has amount of ’65’.
- ‘PRICE3’ has currency of ’AUD’.
- ‘PRICE4’ (AbsolutePrice) has price type of (PriceType) ‘Abs’.
- ‘PRICE4’ has amount of ’50’.
- ‘PRICE4’ has currency of ’AUD’.
- ‘PRICE5’ (AbsolutePrice) has price type of (PriceType) ‘Abs’.
- ‘PRICE5’ has amount of ’25’.
- ‘PRICE5’ has currency of ’AUD’.
- (‘OBG1’, ‘PRICE1’) (ServicePrice) has modifier of ‘Exact’.
- (‘OBG1’, ‘PRICE1’) has price granularity of (ItemGranularity) ‘IG1’.
- (‘OBG1’, ‘PRICE1’) has price granularity of ‘IG2’.
- ‘IG1’ has cardinality of ‘1’.
- ‘IG1’ has granularity of ‘Adult’.
- ‘Adult’ has unit of ‘Person’.
- ‘IG1’ has number of ‘1’.
- ‘IG2’ has cardinality of ‘1’.
- ‘IG2’ has granularity of ‘Ticket’.
- ‘Ticket’ has unit of ‘Permit’.

– ‘IG2’ has number of ‘2’.
– ‘IG2’ relates to (Item) ’Premium Seat’.
– (‘OBG1’, ‘PRICE2’) (ServicePrice) has modifier of ‘Exact’.
– (‘OBG1’, ‘PRICE2’) has price granularity of (ItemGranularity) ‘IG1’.
– (‘OBG1’, ‘PRICE2’) has price granularity of ‘IG3’.
– ‘IG1’ has cardinality of ‘1’.
– ‘IG1’ has granularity of ‘Adult’.
– ‘Adult’ has unit of ‘Person’.
– ‘IG1’ has number of ‘1’.
– ‘IG3’ has cardinality of ‘1’.
– ‘IG3’ has granularity of ‘Ticket’.
– ‘Ticket’ has unit of ‘Permit’.
– ‘IG3’ has number of ‘2’.
– ‘IG3’ relates to (Item) ’A Reserve Seat’.
– (‘OBG1’, ‘PRICE3’) (ServicePrice) has modifier of ‘Exact’.
– (‘OBG1’, ‘PRICE3’) has price granularity of (ItemGranularity) ‘IG1’.
– (‘OBG1’, ‘PRICE3’) has price granularity of ‘IG4’.
– ‘IG1’ has cardinality of ‘1’.
– ‘IG1’ has granularity of ‘Adult’.
– ‘Adult’ has unit of ‘Person’.
– ‘IG1’ has number of ‘1’.
– ‘IG4’ has cardinality of ‘1’.
– ‘IG4’ has granularity of ‘Ticket’.
– ‘Ticket’ has unit of ‘Permit’.
– ‘IG4’ has number of ‘2’.
– ‘IG4’ relates to (Item) ’B Reserve Seat’.
– (‘OBG1’, ‘PRICE4’) (ServicePrice) has modifier of ‘Exact’.
– (‘OBG1’, ‘PRICE4’) has price granularity of (ItemGranularity) ‘IG1’.
– (‘OBG1’, ‘PRICE4’) has price granularity of ‘IG5’.
– ‘IG1’ has cardinality of ‘1’.
– ‘IG1’ has granularity of ‘Adult’.
– ‘Adult’ has unit of ‘Person’.
– ‘IG1’ has number of ‘1’.
– ‘IG5’ has cardinality of ‘1’.
– ‘IG5’ has granularity of ‘Ticket’.
– ‘Ticket’ has unit of ‘Permit’.
– ‘IG5’ has number of ‘2’.
– ‘IG5’ relates to (Item) ’C Reserve Seat’.
– (‘OBG1’, ‘PRICE5’) (ServicePrice) has modifier of ‘Exact’.
– (‘OBG1’, ‘PRICE5’) has price granularity of (ItemGranularity) ‘IG1’.
– (‘OBG1’, ‘PRICE5’) has price granularity of ‘IG6’.
– ‘IG1’ has cardinality of ‘1’.
– ‘IG1’ has granularity of ‘Adult’.
– ‘Adult’ has unit of ‘Person’.
– ‘IG1’ has number of ‘1’.
– ‘IG6’ has cardinality of ‘1’.

- – 'IG6' has granularity of 'Ticket'.
- – 'Ticket' has unit of 'Permit'.
- – 'IG6' has number of '2'.
- – 'IG6' relates to (Item) 'D Reserve Seat'.
- – We enumerate the remaining prices in a manner similar to that for adults. We realise that this requires consensus/understanding on what the granularity of the price (e.g. Adult) actually means.

◇ Payment obligation verbalisations:

- – 'OBG2' (PaymentObligation) has obligation type of 'P'.
- – PaymentObligation 'OBG2' has payment option of (PaymentOption) 'PAYOPT1'.
- – 'PAYOPT1' can be fulfilled using payment instrument 'PAYINST1' at payment location of 'PH1'.
- – 'PAYOPT1' can be fulfilled using payment instrument 'PAYINST2' at payment location of 'PH1'.
- – 'PAYOPT1' can be fulfilled using payment instrument 'PAYINST3' at payment location of 'PH1'.
- – 'PAYOPT1' can be fulfilled using payment instrument 'PAYINST4' at payment location of 'PH1'.
- – 'PAYINST1' (CardBasedInstrument) has payment instrument type of 'C'.
- – 'PAYINST1' is issued by (Provider) 'P2'
- – 'PAYINST1' has card type of 'Charge'.
- – 'PAYINST2' (CardBasedInstrument) has payment instrument type of 'C'.
- – 'PAYINST2' is issued by (Provider) 'P3'
- – 'PAYINST2' has card type of 'Charge'.
- – 'PAYINST3' (CardBasedInstrument) has payment instrument type of 'C'.
- – 'PAYINST3' supports payment scheme name of 'VISA'.
- – 'PAYINST3' has card type of 'Credit'.
- – 'PAYINST4' (CardBasedInstrument) has payment instrument type of 'C'.
- – 'PAYINST3' supports payment scheme name of 'MASTERCARD'.
- – 'PAYINST4' has card type of 'Credit'.

5 Modelling issues

This section discusses issues such as flexibility, recursion and population of the models presented within this paper.

5.1 Model flexibility

The temporal model that we present here is capable of presenting the same information in multiple ways. For example, an entertainment service that is available temporally on Monday through Friday from 9am - 5pm for a specific period can be represented as:

◇ Five distinct temporal intervals that cover each day of the week for the specified period; or

◇ A single temporal interval that covers Monday through Friday with 5 temporal interval restrictions over that period.

This presents a dilemma for the discovery process. To avoid reducing the flexibility of our model we have decided to allow for the representation of time in these multiple ways. We feel it is the responsibility of another service provider to implement the functionality that compares temporal intervals. These temporal interval relations are outlined in detail in [2]. Two options are available to us. Firstly, we can define an interface to the temporal interval functions and allow external parties to provide concrete implementations of those functions. Alternatively, we could find an existing implementation of temporal interval operations, extract an ORM schema that models that implementation and then utilise it. Our preference is for the former approach.

5.2 Recursive nature

By its nature the models presented here are recursive. For example, some services offer a warranty that is actually provided by another service provider. This is catered for in our warranty model in section 3.10.9. By providing a link to the provider who undertakes warranty related issues, we also have the ability to query and discover non-functional properties of that provider. This theoretically would allow us to query for services where the warranty provider has a temporal or locative availability that is appropriate to the service requestor.

5.3 Catalogue pre-population

In the paper we have distinguished the need for parts of the model to be populated by the catalogue provider rather than the service provider. For example, common regions such as states and countries could be defined. This pre-population of the catalogue reduces the likelihood that two service providers would create their own regions for exactly the same entity. This increases the ability to accurately and efficiently filter services during the discovery process, as well as reducing the description effort required of the service provider.

6 Related work

The Universal Description, Discovery and Integration (UDDI) standard supports the notion of a businessEntity (i.e. the service provider), businessServices (i.e. a common grouping of web services), a bindingTemplate (i.e. which outlines the technical details of how to interact with a web service) and a tModel (that represents a unique concept) [29]. The UDDI standard does define some non-functional properties such as addresses, phone numbers, and email addresses for businessEntity contacts. Classifiers are available for all entity types and identifiers are available for businessEntities and tModels. External schemes are used to provide the classification (e.g. UNSPSC) and identification (e.g. Dun and Bradstreet D-U-N-S number). Items such as the classification of relationships between businessEntities are also available. UDDI is focussed on acting as a catalogue for services, particularly web services. It provides a significant set of APIs for retrieving, publishing, subscribing, replicating, and securing catalogue information. The concepts presented in our work could be persisted to UDDI tModels with relative ease.

A recent initiative is the Web Services Modeling Ontology (WSMO) that attempts to provide a framework for use in describing web services, as well as their properties [7]. The framework offers a short outline of the type of non-functional properties that are required: accuracy (error rate), financial, network quality of service, performance, reliability, robustness, scalability, security, transactions and trust. Other properties such as contributor, title, date, description and subject are largely centered around Dublin Core metadata properties [31]. Ontologies are offered for date/time, location and purchasing. The date/time ontology offers support for instants and intervals but fails to support recurring time intervals (like those offered herein) which are regularly used by service providers in the temporal decription of their services. The locative ontology supports notions of country, address, city, state, and borders but we feel lacks semantically powerful locative entities such as URIs, routes, spectra and street directory references. The ontology that they provide for purchasing is a WSML version of the RosettaNet 3A4 PIP.

The Ontology Web Language (OWL) based web service ontology (previously referred to as DAML-S [3]) does provide for the description of a handful of non-functional properties [24]. This initiative is attempting to provide core language constructs for describing services and their properties. The properties include: the service name, a text description, a quality rating, a service category and a placeholder for the inclusion of other non-functional properties of services. The latter is facilitated using the "ServiceParameter" available on the ServiceProfile.

The Web Services Description Language (WSDL) offers only a functional view of web services and does not allow for the description of non-functional properties [8].

The standards that we outline here concentrate on the description of web services. We feel that whilst these standards have promoted the description of electronic services, they have lacked the support for traditional services. We argue that services can be requested electronically, and delivered traditionally. This blurs the lines between the description of web and traditional services. We feel that the description of non-functional service properties must cater for availability, obligations, payment, price, rights, penalties, discounts, quality, trust, security, and reward schemes as they occur so frequently in the services that we interact with on a daily basis.

7 Conclusions

This paper offers an abstract notation for describing the non-functional properties of services such as payment, price, availability, obligations, rights, security, trust, quality, discounts, and penalties. These non-functional properties exhibit constraints over the functionality. A significant challenge during the development of this language has been to determine the domain independent non-functional properties that currently inhibit the discovery of services. We feel that we have presented an alternative for describing the non-functional properties of services that caters for both traditional and web services. The result is an increased capability for service requestors to discover, select and substitute services that are appropriate to their needs. Our challenge to the web services community is to abstract the standards that they are developing to incorporate support for existing "bricks 'n' mortar" services, particularly since the ability to request and be provisioned with a service can occur interchangably between electronic and non-electronic mediums.

We expect that the richer descriptions that are available through the non-functional properties defined in this paper will have an impact on the role of service catalogues. This change will mandate a tighter relationship between the service requestor and the service catalogue to enable enhanced service discovery. Service descriptions are likely to evolve more regularly to ensure their reliable usage. Non-functional properties have been largely ignored by the web services community. We feel that we offer a mechanism for their introduction into the web service related standards.

References

[1] Defense Advanced Research Projects Agency. Internet Protocol, 1981. Available from `http://www.freesoft.org/CIE/RFC/791/index.htm`, accessed on 22-May-2002.

[2] James F. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[3] Anupriya Ankolekar, Mark Burstein, Jerry R. Hobbs, Ora Lassila, David L. Martin, Sheila A. McIlraith, Srini Narayanan, Massimo Paolucci, Terry Payne, Katia Sycara, and Honglei Zeng. DAML-S: Semantic Markup for Web Services. In *Proceedings of the First International Semantic Web Working Symposium (SWWS): Infrastructure and Applications for the Semantic Web*, pages 411–430, Stanford University, California, USA, 2001.

[4] IP Australia. Designs, 2004. Available from `http://www.ipaustralia.gov.au/design/what_index.shtml`, accessed on 16-Nov-2004.

[5] Tim Berners-Lee, Roy T. Fielding, and Larry Masinter. Uniform Resource Identifiers (URI): Generic Syntax, 1998. Available from `http://www.ietf.org/rfc/rfc2396.txt`, accessed on 19-Jun-2001.

[6] A.C Bloesch and Terry Halpin. ConQuer: A Conceptual Query Language. In *15th International Conference on Conceptual Modeling*, volume 1157, pages 121–133. Springer LNCS, 1996.

[7] Jos de Bruijn, Christoph Bussler, Dieter Fensel, Michael Kifer, Jacek Kopecky, Rubn Lara, Eyal Oren, Axel Polleres, and Michael Stollberg. Web Services Modeling Ontology (WSMO) - Working Draft 21st November 2004, 2004. Available from `http://www.wsmo.org/2004/d2/v1.1/20041121/`, accessed on 22-Nov-2004.

[8] Erik Christensen, Greg Meredith, Francisco Curbera, and Sanjiva Weerawarana. Web Services Description Language (WSDL) 1.1. W3C Note, Ariba, Microsoft and IBM Corporation, March 2001. Available from `http://www.w3.org/TR/2001/NOTE-wsdl-20010315`, accessed on 26-Jun-2001.

[9] Roger Clarke. Human Identification in Information Systems: Management Challenges and Public Policy Issues. *Information Technology & People*, 7(4):6 – 37, 1994.

[10] DEC Corporation, Intel Corporation, and Xerox Corporation. The Ethernet: a local area network, data link layer and physical layer specification, 30th September 1980.

[11] Randall Davis, Howard Shrobe, and Peter Szolovits. What is a Knowledge Representation? *AI Magazine*, 14(1):17–33, 1993.

[12] Dun and Bradstreet. Duns Number Information, 2004. Available from `https://eupdate.dnb.com/dunsnumberinfo.html`, accessed on 18-Nov-2004.

[13] ESRI. GIS Dictionary, 2004. Available from `http://support.esri.com/index.cfm?fa=knowledgebase.gisDictionary.gateway`, accessed on 08-Jul-2004.

[14] International Organization for Standardization. ISO 8601:2000, 2000. Available from `http://www.iso.ch`, accessed on 07-Jun-2001.

[15] Terry Halpin. *Information Modeling and Relational Databases : From Conceptual Analysis to Logical Design*. Morgan Kaufmann Publishers, San Francisco, 2001.

[16] Arthur H. M. ter Hofstede. *Information Modelling in Data Intensive Domains*. Doctor of Philosophy, University of Nijmegen, Nijmegen, The Netherlands, 1993.

[17] Robert C. Lewis and Bernard H. Booms. The Marketing Aspects of Service Quality. In L. Berry, G. Shostack, and G. Upah, editors, *Emerging Perspectives in Services Marketing*, pages 99–104. American Marketing Association, Chicago, Illinois, USA, 1983.

[18] Jeffrey K. MacKie-Mason and Kimberly White. Evaluating and Selecting Digital Payment Mechanisms. In Gregory L. Rosston and David Waterman, editors, *Interconnection and the Internet. Selected Papers from the 1996 Telecommunications Policy Research Conference*, pages 113–134. Lawrence Erlbaum, Mahwah, New Jersey, USA, 1997.

[19] Michelin. The Michelin Guide France, 2004. Available from `http://www.viamichelin.com`, accessed on 10-Aug-2004.

[20] Anthony Nadalin, Chris Kaler, Phillip Hallem-Baker, and Ronald Monzillo. Web Services Security: SOAP Message Security 1.0 (WS-Security 2004), 2003. Available from `http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf`, accessed on 13-Jul-2004.

[21] United Nations. United Nations Standard Products and Services Codes (UN/SPSC), 2000. Available from `http://www.unspsc.org/`, accessed on 15-Nov-2000.

[22] Phillipa Oaks, Arthur H. M. ter Hofstede, and David Edmond. Capabilities: Describing What Services Can Do. In Maria E. Orlowska, Sanjiva Weerawarana, Mike P. Papazoglou, and Jian Yang, editors, *First International Conference on Service-Oriented Computing (ICSOC)*, volume 2910 of *Lecture Notes in Computer Science*, pages 1–16, Trento, Italy, 2003. Springer.

[23] Justin O'Sullivan, David Edmond, and Arthur ter Hofstede. What's in a service?: Towards accurate description of non-functional service properties. *Distributed and Parallel Databases Journal - Special Issue on E-Services*, 12(2-3):117–133, 2002.

[24] OWL-S Coalition. OWL-S Web Service Ontology, 2004. Available from `http://www.daml.org/services/owl-s/1.1/`, accessed on 21-Nov-2004.

[25] United States Patent and Trademark Office. General Information Concerning Patents, 2004. Available from `http://www.uspto.gov/web/offices/doc/general/index.htm`, accessed on 16-Nov-2004.

[26] United States Postal Service. Postal Addressing Standards: Publication 28, November 2000 2000. Available from `http://pe.usps.gov/cpim/ftp/pubs/Pub28/pub28.pdf`, accessed on 08-Jul-2004.

[27] TC37/SC2-TC46/SC4 Joint Working Group. ISO 639-2, Codes for the representation of names of languages–Part 2: Alpha-3 code, 1998. Available from `http://www.loc.gov/standards/iso639-2/`, accessed on 11-Aug-2004.

[28] Steffen Thorsen. Time zone abbreviations, 2004. Available from `http://www.timeanddate.com/library/abbreviations/timezones/`, accessed on 08-Jul-2004.

[29] UDDI.org. UDDI Version 3.0.2 API Specification, July 2004. Available from `http://uddi.org/pubs/uddi-v3.0.2-published-20041019.pdf`, accessed on 17-Jan-2005.

[30] Princeton University. WordNet - a lexical database for the English language. `http://www.cogsci.princeton.edu/cgi-bin/webwn` (17/01/2005).

[31] Stuart Weibel, John Kunze, Carl Lagoze, and Misha Wolf. RFC2413 - Dublin Core Metadata for Resource Discovery, 1998. Available from `http://rfc2413.x42.com/`, accessed on 21-Nov-2004.

[32] Wikipedia. RFID, 2004. Available from `http://en.wikipedia.org/wiki/RFID`, accessed on 16-Nov-2004.

[33] World Intellectual Property Organization (WIPO). *The International Nice Classification of Goods and Services for the Purposes of the Registration of Marks*. World Intellectual Property Organization, 1977.

[34] World Intellectual Property Organization (WIPO). *Establishing an International Classification of the Figurative Elements of Marks*. World Intellectual Property Organization, 1985.

[35] World Intellectual Property Organization (WIPO). *Intellectual Patent Classification (IPC) Complete Set.* World Intellectual Property Organization, 7 edition, 1999.

[36] World Intellectual Property Organization (WIPO). *International Classification for Industrial Designs (Locarno Classification).* World Intellectual Property Organization, 8 edition, 2003.

[37] Valarie A. Zeithaml, A. Parasuraman, and Arvind Malhotra. A Conceptual Framework for Understanding e-Service Quality: Implications for Future Research and Managerial Practice. Research Report 00-04, Kenan Flagler Business School, University of North Carolina, Chapel Hill, North Carolina, 2001. Available from `http://www.kenanflagler.unc.edu/Marketing/wpapers/papers/e-Service.pdf`, accessed on 10-Jul-2001.