

member just to the next level (that is, A has B as direct aggregated member).

Finally, the class "AggregatedObject" has been defined as equivalent to those things that have some values for the property "hasAggregatedMember". This modelling allows the automatic classification of aggregated objects in this class when a reasoner is applied.

2.3 Consequences

This content pattern allows designers to represent both simple individuals of a given concept (that is, an individual that is made up of itself) and aggregated individuals of a given concept (that is, an individual that is made up of several individuals of the same concept). In summary, this pattern allows to represent both simple objects and aggregated objects and their members.

In addition, this pattern can be used to detect the following contradictory situation by means of applying a reasoner: 'to instantiate the relationship "hasAggregatedMember" for an Object that belongs to "SimpleObject"'. This situation represents a consistency error and it is detected when a reasoner is applied due to the following modelling decisions included in the pattern: (a) "AggregatedObject" class represents the "hasAggregatedMember" domain and (b) "AggregatedObject" is disjoint with "SimpleObject".

3. PATTERN USAGE EXAMPLE

This pattern has been applied to different domains such as service providers and context sources during the mIO! ontology network¹ development.

As an example, we show in Figure 2 the application of the SimpleOrAggregated pattern to represent that a service provider can be classified as simple or aggregated. Each service provider can be also classified with respect to the type of service it provides (e.g. cultural, entertainment, food, health, etc.).

4. Related work

The origin of this pattern is the modelling of service providers and context sources into the mIO! ontology network [2] within the Spanish project mIO!². The pattern has been also applied to computing and storage resources modelling in the Metascheduler ontology³ in the context of the Spanish project *España Virtual*⁴.

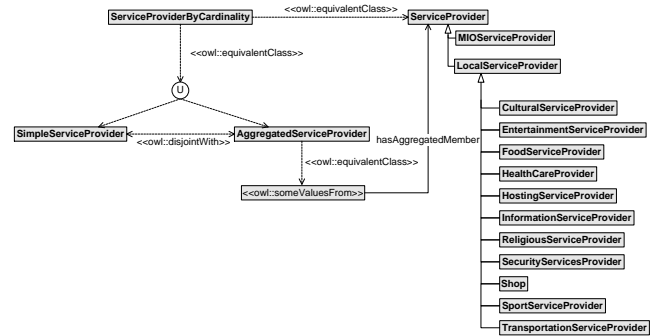


Figure 2. SimpleOrAggregated pattern applied to service providers.

5. Summary and Outlook

The *SimpleOrAggregated* pattern provides a mechanism to classify objects as simple or aggregated objects depending on whether they are an aggregation of some objects. This classification is compatible with another possible classification of objects.

6. ACKNOWLEDGMENTS

This work has been partially supported by the Spanish project *mIO!* (CENIT-2008-1019).

7. REFERENCES

- [1] Presutti, V., Gangemi, A., David S., Aguado de Cea, G., Suárez-Figueroa, M.C., Montiel-Ponsoda, E., Poveda, M. *NeOn D2.5.1: A Library of Ontology Design Patterns: reusable solutions for collaborative design of networked ontologies*. NeOn project. <http://www.neon-project.org>. 2008.
- [2] Poveda, M., Suárez-Figueroa, M.C., García-Castro, R., Gómez-Pérez, A. *A Context Ontology for Mobile Environments*. Proceedings of CIAO 2010. Lisbon, Portugal. 11 October 2010.
- [3] Suárez-Figueroa, M.C., Brockmans, S., Gangemi, A., Gómez-Pérez, A., Lehmann, J., Lewen, H., Presutti, V., Sabou, M. *NeOn D5.1.1: NeOn Modelling Components*. NeOn project. <http://www.neon-project.org>. March 2007.

¹<http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/ontologies/82-mio-ontologies>

² <http://www.cenitmio.es/>

³<http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/ontologies/85-metascheduler-ontologies>

⁴ <http://www.españavirtual.org/>