Distributed Domain Model for the Case-Based Retrieval of Architectural Building Designs

Viktor Ayzenshtadt[1,3], Christoph Langenhan[4], Saqib Bukhari[3]
Klaus-Dieter Althoff[1,3], Frank Petzold[4], Andreas Dengel[2,3]

[1]University of Hildesheim, Institute of Computer Science
[2]Technical University of Kaiserslautern
[3]German Research Center for Artificial Intelligence
Trippstadter Strasse 122, 67663 Kaiserslautern, Germany
{Viktor.Ayzenshtadt, Saqib.Bukhari, Klaus-Dieter.Althoff, Andreas.Dengel}@dfki.de

[4]Chair of Architectural Informatics
Faculty of Architecture Technical University of Munich
Arcisstrasse 21, 80333 Munich, Germany
{Langenhan, Petzold}@ai.ar.tum.de

Abstract In this paper we present a domain model for the attribute-value-based representation of architectural building designs. The model is currently part of the retrieval system prototype MetisCBR, a multi-agent system for the distributed search of architectural designs in case bases that contain these designs. Using the idea of semantic fingerprint of architectural design, we transferred the fingerprint structure and the corresponding architectural specification of the graph markup language (GraphML) to a CBR domain model for implementation in distributed case-based retrieval projects. We extended the model by adding new attributes for more efficiency, and evaluated it using an exemplary user query and a special retrieval strategy to show its advanced development stage and suitability for such retrieval applications.

Keywords: case-based retrieval, case-based design, CBR modeling

1 Introduction

During the early phase of architectural conceptualization process an architect often comes across a situation when she needs inspiration, comparable examples or new ideas for the building design that is currently being developed. A knowledge-based retrieval system that is able to search for architectural designs similar to a created one can be of considerable help in such situations. For the efficient and structured achievement of helpful retrieval results, an underlying knowledge base domain model that takes into account relevant aspects of an architectural building design is one of the essential requirements of the retrieval system. Moreover, it should also be possible to modify and extend the model. For example, to revise or add features that can improve retrieval performance and results immediately as well as in the long term.

In this paper we propose a distributed domain model structure for implementation in distributed case-based retrieval systems, where concurrently working system entities (agents) are responsible for the retrieval of the most similar cases (architectural building designs). We think that this model can be of common interest for the entire *CAAD* (computer-aided architectural design) domain. This paper is also an introduction and a pre-study to a series of user studies (see Section 5).

This paper is organized as follows: at first the related work will be presented – this section consists of short descriptions of research work that is related to the purpose of our model, and was used in other CBD (Case-Based Design) retrieval systems, as well as of the description of the basic research project *Metis* in context of which the retrieval system MetisCBR and the proposed domain model were developed. In the next section we give a detailed description of the model itself, including the semantic building description model *Semantic Fingerprint* and the corresponding specification of the architectural GraphML *(AGraphML)*, description of the concepts of the model (including their corresponding attributes, similarity measures, and hierarchical relations), and a special retrieval strategy. Finally, we present the evaluation of the model with an exemplary user query. The conclusion summarizes the paper and provides a short description of the future work.

2 Related Work

In this section we describe the work that is closely related to our retrieval model. We distinguish between the related work that was carried out in context of the *Metis* research project and external approaches of other systems and projects.

2.1 Project-related Work

The model we propose in this paper is part of the basic research project *Metis – Knowledge-based search and query methods for the development of semantic information models (BIM) for use in early design phases* [13][1], an interdisciplinary joint project of the German Research Center for Artificial Intelligence *(DFKI)* and the Technical University of Munich *(TUM)*. The project is partially funded by the German Research Foundation *(DFG)*. The interdisciplinarity of the project allows for combination of the aspects of the following areas: CBR, Multi-Agent Systems *(MAS)*, CAAD, and BIM (Building Information Modeling).

- CBR, as technique that works with retained knowledge and experience, is an applied method for determination of similar cases (building designs) inside a case base of such designs.
- MAS (the system agents) is the part of the project infrastructure that executes the actual retrieval by applying CBR methods and special retrieval strategies.
- CAAD, being the closest architecture-related area, provides insights of the architectural view on retrieval to help to achieve the most relevant results.
- BIM (being part of CAAD) provides the currently most well-known basic structure for architectural semantic information models.

Within the ongoing project activities some substantial research and development work has been done to date. The most up-to-date work is the web-based user interface [4] for construction and transfer of user queries to the MetisCBR retrieval system. This user interface is able to build queries in the AGraphML format [15] that is based on the *architectural* specification of GraphML [5]. Furthermore, other interfaces for query construction exist: an Android application [12], a touch-table interface [14], and an iPad application.

Besides the case-based retrieval approach, where our presented model serves as a basis, other retrieval approaches were developed for the project aims as well. For example, a subgraph matching-based algorithm, that includes processing and information extraction from the graphical floor plan queries in form of sketches, was presented in [1]. A *semantic fingerprint* structure, that is intended to hierarchically describe an abstracted representation of a building design, was proposed in [16] (see also Section 3 for more detailed description). The fingerprint structure is used as a basis in the mentioned subgraph-matching algorithm and in the index-based retrieval approach that is currently being developed for the project.

To unite these approaches in a common retrieval system, a distributed retrieval approach was initiated, where a research work of Siebert [22] provides an initial setting based on the SEASALT architecture [19]. A research work of Ayzenshtadt [2] completes and extends this approach by adding new organizational mechanisms, new agents and agent groups, case representations, and a communication ontology. The domain model described in this paper is also part of [2].

2.2 External Related Work

In the last decades a number of different approaches were proposed to introduce and adapt case-based reasoning techniques for the conceptualization process of architectural design. Overviews of these approaches can be found in [9] and [21]. One of the most current publications that contains an overview of case-based reasoning in the architectural domain as well as comprehensive and detailed descriptions of architecture-related CBR approaches is [20].

One of the approaches most related to our retrieval approach is FABEL [24], a multifunctional system with so-called *specialists* that use an aspect-based representation of cases to determine the

[1] Metis – *Wissensbasierte Such- und Abfragemethoden für die Erschließung von Informationen in semantischen Modellen (BIM) für die Recherche in frühen Entwurfsphasen.*

most similar ones to a given query, that is also transformed to the aspect-based representation during the retrieval process. The case base of aspect-based designs can be represented as network of cube-based case representations where each side of a cube is an aspect of the corresponding case. If two cases possess an identical aspect they can be connected through a relational arc (see also Figure 1).

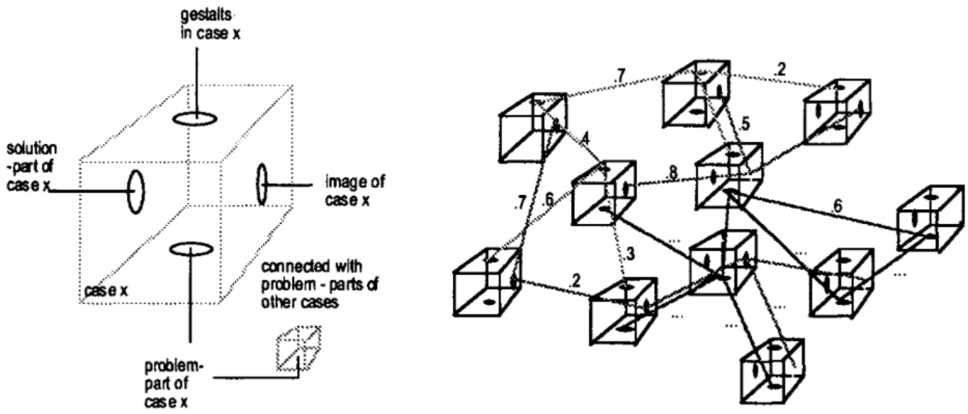

Figure 1. Case base ASPECT in FABEL (Source: [24]).

A different, but related approach is CBArch [6], a system for support of the architectural conceptualization phase that is concentrated on buildings with commercial background. The approach includes all steps of the 4R CBR-cycle (Retrieve, Reuse, Revise, Retain), and uses an ontology for representation of building shapes. The ontology increases its scope with addition of new designs.

To complete the list of external related work with comparable purposes, other approaches, including CaseBook [11], PRECEDENTS [18], SEED [7] or DYNAMO [10], can also be named.

3 Semantic fingerprint-based domain model

In this section we present our distributed CBR domain model for the retrieval of similar building designs, including its underlying structure, concepts, relations, properties, attributes, similarity measures, and a special retrieval strategy.

3.1 Semantic fingerprint

Semantic fingerprint [16] is a hierarchical, index-based structure for representation of meta data of floor plans. This structure can be used for description or querying within a retrieval system to find similar architectural projects. Besides the semantic information, topological properties of architecture are also used to create a fingerprint. The fingerprint structure divides a building construction into four hierarchically ordered main concepts: *Levels* that contain *Units* that are divided into *Zones* that consist of particular *Rooms*. For each of the concepts, entities exist that describe it in detail (for example, a zone can be a *Living* or a *Sleeping* zone). Direct or adjacent relations connect the different floor plan spaces to each other.

3.2 AGraphML specification

The specification [15] of GraphML [5] for the architectural domain is the underlying structure for representation of building designs by means of applying an XML-based markup. In the *Metis* project this representation is the common intermediate representation of *cases*, *queries*, and *semantic fingerprints*, and is the exchange format that is used by the web-based user interface [4] and the retrieval system MetisCBR.

The AGraphML specification is built hierarchically and consists of three main concepts that represent attribute-value-based information about a particular building design: *Graph* (meta data of a floor plan) includes *Nodes* (information about rooms) that are connected with *Edges* (room connections). The corresponding attributes describe the main concepts in detail (see more about attributes in the Section 3.4)

3.3 Model structure

The general model setting is based on the above mentioned *semantic fingerprint* structure and the corresponding AGraphML specification. Thus, we use the three main concepts of AGraphML (*Graph meta data*, *Node*, and *Edge*) and their corresponding attributes as a basis, and build our domain model upon them (for more understandable structure of the model, two renaming transformations were conducted: *Graph meta data* $\rightarrow$ *Floorplan* and *Node* $\rightarrow$ *Room*). Another substantial reason for using the AGraphML specification as a template for the domain is the format of user queries of the web-based user interface (see Section 2.1), which is also AGraphML. For modeling we use the *myCBR* framework [3] (Version *3.1 beta*). The general setting of the model can be seen in Figure 2.

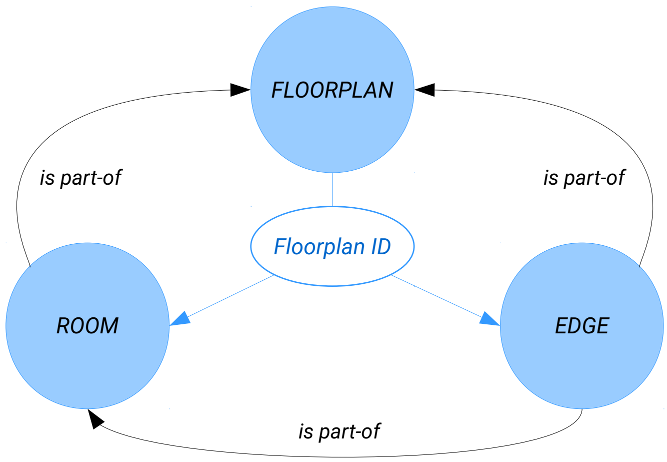

Figure 2. The general structure of the domain model.

The direct connection between these three main concepts is established with *is part-of* relations. An *Edge* can be seen as part of both *Floorplan* and *Room*, whereas *Room* itself is only connected to *Floorplan* through this kind of relation. An indirect connection exists by means of applying the unique ID of the common floor plan as additional attribute to every *Room* and *Edge* (see also Section 3.5).

3.4 Attributes

For each of the main concepts of the domain model, the corresponding attributes of the AGraphML specification were transferred to provide the complete compatibility of the model with user queries in AGraphML format. The Tables 1, 2 and 3 show the attributes of the main concepts, including the type and a short description for each attribute.

Attribute	Type	Description
imageUri	string	URI of the building design
imageMD5	string	MD5 check sum of the building design
ifcUri	string	URI of the IFC (Industry Foundation Classes) server
buildingId	string	GUID of the building
id	string	GUID of the floor plan
bimserverPoid	float	bimserverId of the IFC source
validatedManually	boolean	Has the design been manually corrected or improved?
alignmentNorth	float	Alignment of the design in the direction of North
geoReference	string	Geographic coordinates
scaleIsMeter	boolean	Is scale factor meter?
scaleToMeter	float	Scale factor to scale to meter (1.0 = scale is meter)
floorLevel	float	Floor level of the floor plan

Table 1. *Floorplan* attributes.

Attribute	Type	Description
id	string	Either IFC GUID (IfcSpaces) or generated internally
roomType	symbol	Room type (defined in a list, see below)
name	string	Room name (defined by the user)
center	string	Room center point as WKT (Well-known text) point
corners	string	Room corners as WKT polygon
windowExist	boolean	Does the room have windows?
enclosedRoom	boolean	Is the room completely covered by walls?
light	integer	Room luminance (scale 1-20)
privacy	integer	Room privacy factor (scale 1-20)
area	float	Room area (defined by user or computed from corners data)
zone	symbol	Which zones (see Section 3.1) does the room belong to?

Table 2. *Room* attributes.

For the attribute `roomType`, following room types are defined at the moment: *Room, Kitchen, Living, Sleeping, Working, Corridor, Toilet, Bath, Exterior, Storage, Buildingservices, Children, Parking.*

Attribute	Type	Description
id	string	IFC GUID (IfcWall, IfcDoor etc.) or generated internally
edgeType	symbol	Room connection type (defined in a list, see below)
weight	float	Edge weight
linearDistance	float	Direct linear distance of room center points
position	string	Position of doors or windows as WKT point
walkingDistance	float	Distance of room center points
feltDistance	float	Perceived distance between rooms
viewRelation	integer	View relation from one room to another (scale 1-20)

Table 3. *Edge* (room connection) attributes.

For the attribute `edgeType`, following room connection types are defined at the moment: *Door, Entrance, Passage, Slab, Stairs, Wall, Window.*

3.5 Additional Attributes

In addition to the existing AGraphML attributes we added some attributes for more efficiency in the retrieval and more plausible structure of the domain model. These attributes are listed in the Tables 4, 5 and 6.

Attribute	Type	Description
ROOM	Concept	A room that is contained in this floor plan (multiple additions are possible). Added to assure the *is part-of* relation.
`roomCount`	`integer`	Number of rooms in the floor plan. Added for usage in retrieval strategies.
`roomTypes`	`string`	A list of room types contained in the floor plan. Added for usage in retrieval strategies.

Table 4. Additional attributes of *Floorplan*.

Attribute	Type	Description
EDGE	Concept	A room connection that starts or ends in this room (multiple additions are possible). Added to assure the *is part-of* relation.
`floorplanId`	`string`	ID of the floor plan that the room belongs to. Added for usage in retrieval strategies.

Table 5. Additional attributes of *Room*.

Attribute	Type	Description
`source`	`symbol`	Room where the connection starts. Added to make the edge similarity assessment more efficient by comparing the source and target of an edge as well.
`target`	`symbol`	Room where the connection ends. Added to make the edge similarity assessment more efficient by comparing the source and target of an edge as well.
`floorplanId`	`string`	ID of the floor plan that the room belongs to. Added for usage in retrieval strategies.

Table 6. Additional attributes of *Edge*.

3.6 Similarity measures

To assure the efficiency of the case-based retrieval, and to provide results that can be considered helpful for the user, a set of similarity measures is required that is adapted to the properties of the corresponding model.

The similarity assessment of our model follows the organization of the similarity computation implemented in *myCBR*: local similarity measures on the *attribute level* and a global weighted amalgamation measure on the *concept level*.

For each of the model's main concepts, we determined two amalgamation functions: a *default* one with no weighting on the global level, and an *advanced* one with attribute weighting according to its relevance in the concept. For both amalgamation measures we have selected the *Euclidean Norm* as underlying function.

On the local level we defined a similarity measure for each of the attributes according to its attribute type, for each of the amalgamation measures:

- a *default* local similarity that is associated with the *default* amalgamation
- an *advanced* similarity measure (e.g., *ordered function* for `symbol` attributes) that is associated with the *advanced* amalgamation

We also assigned *exemplary* weights (scale 1-10) to some attributes for the *advanced* amalgamation. The most notable results are the following:

– *Floorplan:* the attributes `buildingId`, `floorLevel`, `roomCount` and `roomTypes` got a high weight value, that is, are considered key attributes
– *Room:* the attribute `roomType` is not considered relevant, as a room can be assigned with a different function by transformation of the interior
– *Edge:* the attributes `source` and `target` are the key attributes and got a high weight value of 8.0

In Figure 3, an example of a local function is depicted. For the complete overview of all similarity measures, a myCBR project (that was also used for the evaluation of the model, see Section 4) can be requested by contacting the first author of this paper.

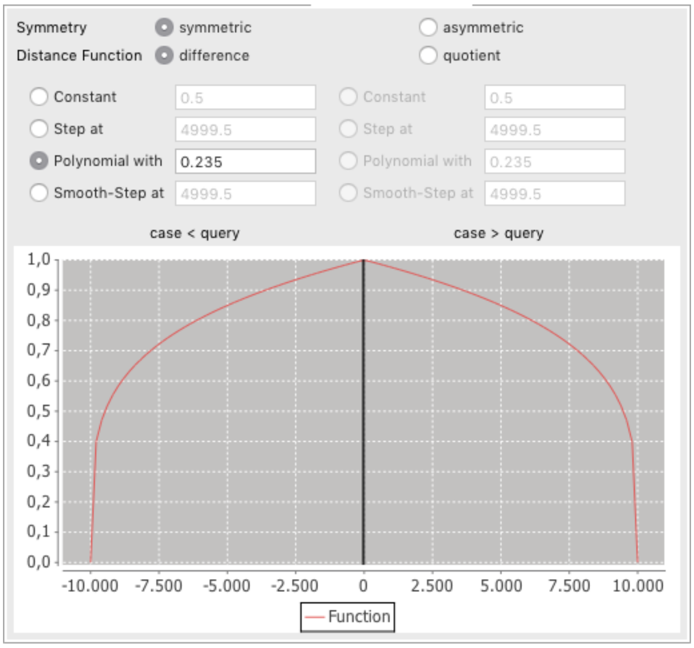

Figure 3. The local similarity function of the `area` attribute (concept *Room*).

3.7 Fingerprint derivation

The main aim of the domain model – to find the most similar cases in the given case base(s) – is also possible with derivation of particular semantic fingerprints as functions, that can act as more refined and adapted patterns for the search of helpful designs. For the *Metis* project, a list of such fingerprints was defined. These fingerprints are used for both similarity assessment, and to help the user to refine the search by selecting one or many fingerprints during the retrieval process.

A single fingerprint pattern is a graph-based representation and, as well as our model, is based on the *semantic fingerprint* structure and the AGraphML specification. Every fingerprint pattern is characterized by the corresponding description label and properties, that indicate how the graph components (rooms and room connections) are connected and labeled, or attributes from which concept(s) will be used to determine similarity. For example, a combination of the description label *Natural Light* and the property *Node attribute* indicates a fingerprint that is intended to be used for identification of cases where existence of windows in exterior walls plays the most significant role.

For our domain model, it is currently planned to implement these fingerprint patterns/derivations as particular amalgamation functions, as it is usual for a fingerprint representation to combine a number of different attributes in one fingerprint. We see this functionality as related to *footprint sets* described in [23].

3.8 Retrieval strategy

Closely related to the structure, concepts, attributes, and the similarity functions of the model is the retrieval strategy, that is applied to the user query during the retrieval process. In MetisCBR, retrieval strategies determine how a particular retrieval process will be accomplished, and which retrieval agents will be involved in this particular process. Currently, three types of retrieval agents exist in our retrieval system MetisCBR: *Floorplan agent*, *Room agent*, and *Edge agent*. As their names suggest, each of these agents is associated with the model's main concept of the same name. During the retrieval process they are supported by a retrieval manager agent. At the current stage of development the most advanced and implemented special retrieval strategy is the *Basic* strategy that is inspired by the MAC/FAC method [8].

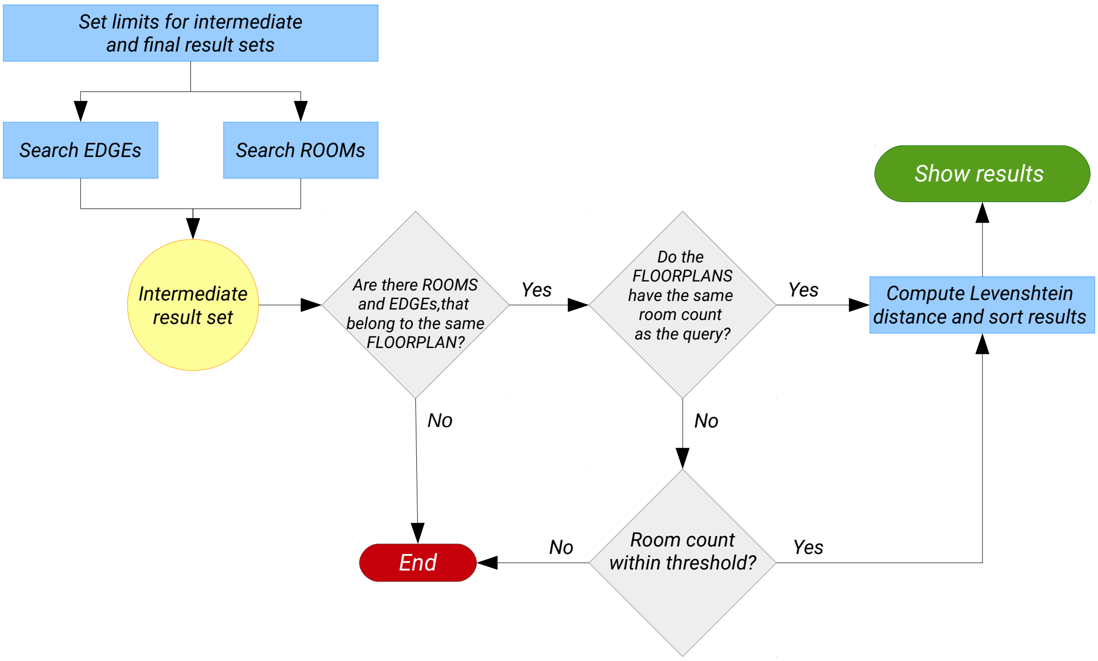

Figure 4. The *Basic* strategy.

In the basic strategy (see Figure 4), first the limits of the final and the intermediate result set are determined. In the second step, the *Room* and *Edge* agents search for the most similar rooms and room connections in the case base(s) (with attributes according to the amalgamation measure) and build an intermediate result set with capacity according to the limits. Rooms and room connections contained in this intermediate set that have the same value of the `floorplanId` attribute are considered *matched*, the floor plan ID will be then placed on the new result list. If there are no matching elements found, the retrieval process is terminated.

In the next step, we compare the `roomCount` of each floor plan from the ID result list with the number of the rooms in the user query. If the room count is not equal to the query's room count, or is not within a tolerable threshold value, then it is removed from the result list. If after this step no floor plan with proper room count value is left, then the retrieval process is terminated.

In the last step of the *Basic* strategy, we sort the floor plans according to the similarity of their `roomTypes` attribute to the same attribute of the user query. To sort, we currently use a Levenshtein distance-based string similarity measure described in [17].

4 Evaluation

The evaluation of the model was intended to provide a proof of its suitability for CBD/CAAD retrieval applications that use case-based retrieval as the underlying search method, and to assure the model's legitimation for usage in our next user studies.

For this evaluation process, we used the building design collection of mediaTUM[2] – a media content management system of the TU Munich. We imported 1477 instances of the main concepts *Floorplan*, *Room* and *Edge* into the case base, that has our model as its underlying structure, where 112 *Floorplans* stand for the number of retrievable building designs.

We used a query in AGraphML format, that consists of 5 rooms and 6 room connections (see Table 7), and queried the retrieval system with 2 different requests, using each of the created amalgamation functions (see Section 3.6) with the special *Basic* retrieval strategy (see Section 3.8).

Rooms	**Edges**	**Room types**	**Edge types**
5	6	CORRIDOR-1 (C1) CORRIDOR-2 (C2) SLEEPING (S) STORAGE (ST) BATH (B)	PASSAGE-1 (S1->S2) PASSAGE-2 (ST->C2) PASSAGE-3 (B->C2) PASSAGE-4 (ST->B) DOOR (C1->C2) WALL (ST->C2)

Table 7. Properties of the query for the model evaluation.

To rate the quality of the results, we used the similarity value that is computed in the last step of the *Basic* strategy. We classified the results into *very similar*, *similar*, *sufficiently similar* and *unsimilar*. As very similar we considered result cases that have the similarity $Sim \geq 0.75$. For similar results $0.75 > Sim \geq 0.5$. For sufficiently similar results $0.5 > Sim \geq 0.25$. For unsimilar results $Sim < 0.25$. For this evaluation we set the threshold for the final result set to 1, that is, $4 \leq roomCountFromCase \leq 6$ (Table 8 shows the quantities of the relevant cases in the case base). We set the limit for the number of retrieved intermediate results to 20 for each *Room* and *Edge* search request to ensure that the amalgamation functions don't work with the same intermediate result set. We did not set the limit for the final result set.

Total number of cases	$roomCount = 6$	$roomCount = 5$	$roomCount = 4$
112	27	24	23

Table 8. Quantities of relevant cases in the case base.

The results of the evaluation have confirmed, that the model, in combination with the strategy and amalgamation functions, is able to retrieve cases that can be considered useful, if the room count value will be used as the main relevance criterion (we were aware of the fact that there is no definite assessment of quality of results in design retrieval, as every architect has her or his own rating criteria). Considering designs that consist of 5 rooms as the most relevant cases, the *advanced* (weighted) amalgamation has delivered better (and in general slightly more) results than the *default* amalgamation (see Tables 9 and 10).

[2] https://mediatum.ub.tum.de/

Total	$roomCount = 6$	$roomCount = 5$	$roomCount = 4$
23	14	7	2

Table 9. Result quantities of the search with the *default* unweighted amalgamation.

Total	$roomCount = 6$	$roomCount = 5$	$roomCount = 4$
27	11	12	4

Table 10. Result quantities of the search with the *advanced* weighted amalgamation.

The rating with similarity distribution showed, as expected, better results for the *advanced* amalgamation. The Tables 11 and 12 show the results of similarity distribution of the amalgamation functions. We believe that weighting of attributes also is a process where every architect will have an own method and criteria for assignment of the weights. Therefore, these results should not be seen as universally valid (see the notice about the relevance criteria in the previous paragraph). Nevertheless, attribute weighting is required to get higher similarity values and more relevant result cases. Increasing the number of results in the intermediate set of the *basic* strategy will also bring changes for both quantity of the final result set and the corresponding similarity distribution.

Highest Sim	$\varnothing Sim$	$Sim \geq 0.75$	$0.75 > Sim \geq 0.5$	$0.5 > Sim \geq 0.25$	$Sim < 0.25$
0.65	≈ 0.41	0	5	18	0

Table 11. Similarity distribution with the *default* unweighted amalgamation.

Highest Sim	$\varnothing Sim$	$Sim \geq 0.75$	$0.75 > Sim \geq 0.5$	$0.5 > Sim \geq 0.25$	$Sim < 0.25$
0.825	≈ 0.44	1	7	19	0

Table 12. Similarity distribution with the *advanced* weighted amalgamation.

5 Conclusion and Future Work

In this paper we presented a CBR domain model for distributed retrieval of architectural designs. The model can be used within a retrieval system, such as MetisCBR, a multi-agent system of the *Metis* project that is currently under active development and aims to support the conceptualization phase in architecture. We presented the underlying *semantic fingerprint* and AGraphML structures that are the theoretical basis of our model. We also presented the organization of the domain model, its main concepts, attributes and additional attributes, together with similarity measures, and the retrieval strategy. The approach was evaluated with an exemplary AGraphML query for the purpose of demonstration of the advanced stage of the model development and its appropriateness for the implementation in retrieval systems.

This paper was intended to play a role of introduction to the series of user studies that we are going to conduct in the nearest future. In these studies we will compare approaches of the retrieval of architectural designs, including case-based, index-based and rule-based systems. Before the beginning of the studies these systems will be improved, for example, by implementing fingerprint derivations in our presented model. This development step will bring consistency for the approaches, and ensure that each of them has the same preconditions. Based on the evaluation results presented in this paper, we will use the weighted amalgamation measure for the studies.

6 Acknowledgement

This work was supported by the DFG (Deutsche Forschungsgemeinschaft, *German Research Foundation*).

References

1. Ahmed, S., Weber, M., Liwicki, M., Langenhan, C., Dengel, A., Petzold, F.: Automatic analysis and sketch-based retrieval of architectural floor plans. Pattern Recognition Letters 35, 91–100 (2014)
2. Ayzenshtadt, V.: Concept and Implementation of a Distributed Case-based Retrieval Approach for the Support of Architectural Conceptualization Phase. Master thesis. University of Hildesheim (2015)
3. Bach, K., Althoff, K.D.: Developing case-based reasoning applications using myCBR 3. In: Watson, I., Diaz Agudo, B. (eds.) Case-based Reasoning in Research and Development. International Conference on Case-Based Reasoning (ICCBR-2012), 20th, September 3-6, Lyon, France, vol. 7466, pp. 17–31, LNCS. Springer-Verlag, Berlin Heidelberg (2012)
4. Bayer, J., Bukhari, S., Dengel, A., Langenhan, C., Althoff, K.D., Petzold, F., Eichenberger-Liwicki, M.: Migrating the classical pen-and-paper based conceptual sketching of architecture plans towards computer tools - prototype design and evaluation. 11th IAPR International Workshop on Graphics Recognition - GREC'15, Nancy, France (2015)
5. Brandes, U., Eiglsperger, M., Herman, I., Himsolt, M., Marshall, M.S.: Graphml progress report structural layer proposal. In: Graph Drawing. pp. 501–512. Springer (2002)
6. Cavieres, A., Bhatia, U., Joshi, P., Zhao, F., Ram, A.: CBArch: A Case-Based Reasoning Framework for Conceptual Design of Commercial Buildings. Artificial Intelligence and Sustainable Design – Papers from the AAAI 2011 Spring Symposium (SS-11-02) pp. 19–25 (2011)
7. Flemming, U.: Case-based design in the SEED system. Automation in Construction 3(2), 123–133 (1994)
8. Gentner, D., Forbus, K.D.: MAC/FAC: A model of similarity-based retrieval. In: Proceedings of the thirteenth annual conference of the Cognitive Science Society. vol. 504, p. 509. Citeseer (1991)
9. Heylighen, A., Neuckermans, H.: A case base of case-based design tools for architecture. Computer-Aided Design 33(14), 1111–1122 (2001)
10. Heylighen, A., Schreurs, J., Neuckermans, H.: Enter instead of submit. DesignNet-Knowledge e Information Management per il design pp. 171–182 (2002)
11. Inanc, B.S.: Casebook. an information retrieval system for housing floor plans. In: the Proceedings of 5th Conference on Computer Aided Architectural Design Research (CAADRIA). pp. 389–398 (2000)
12. KSD Research Group: ar:searchDroid. `http://ksd.ai.ar.tum.de/?project=arsearchdroid-3&lang=en` (2015)
13. KSD Research Group: KSD - Forschungsprojekte - metis. `http://ksd.ai.ar.tum.de/?page_id=140&lang=en` (2015)
14. KSD Research Group: Touchtect 2.0 and Unified Query Service. `http://ksd.ai.ar.tum.de/?project=809&lang=en` (2015)
15. Langenhan, C.: A federated information system for the support of topological bim-based approaches. Forum Bauinformatik Aachen (2015)
16. Langenhan, C., Petzold, F.: The fingerprint of architecture-sketch-based design methods for researching building layouts through the semantic fingerprinting of floor plans. International electronic scientific-educational journal: Architecture and Modern Information Technologies 4, 13 (2010)
17. Naumann, F: Similarity Measures. Hasso-Plattner Institut. Data Profiling and Data Cleansing. `http://hpi.de/fileadmin/hpi/FG_Naumann/folien/SS13/DPDC/DPDC_12_Similarity.pdf` (2013)
18. Oxman, R., Oxman, R.: Precedents: Memory structure in design case libraries. In: CAAD Futures. vol. 93, pp. 273–287 (1993)
19. Reichle, M., Bach, K., Althoff, K.D.: The SEASALT architecture and its realization within the docquery project. In: KI 2009: Advances in Artificial Intelligence, pp. 556–563. Springer (2009)
20. Richter, K.: Augmenting Designers' Memory: Case-based Reasoning in Architecture. Logos-Verlag (2011)
21. Richter, K., Heylighen, A., Donath, D.: Looking back to the future – an updated case base of case-based design tools for architecture. Knowledge Modelling-eCAADe (2007)
22. Siebert, M.: Concept of aCBR System for the Architectural Conceptualization Phase Support. Project report. (2014)
23. Smyth, B., McKenna, E.: Footprint-based retrieval. In: Case-Based Reasoning Research and Development, pp. 343–357. Springer (1999)
24. Voss, A.: Case design specialists in FABEL. Issues and Applications of Case-based Reasoning in Design pp. 301–335 (1997)

All links were last followed on Oct 15, 2015.