

Comparative Evaluation of Rule-Based and Case-Based Retrieval Coordination for Search of Architectural Building Designs

Viktor Ayzenshtadt^{1,3}, Christoph Langenhan⁴, Johannes Roith⁴, Saqib Bukhari³, Klaus-Dieter Althoff^{1,3}, Frank Petzold⁴, Andreas Dengel^{2,3}

¹University of Hildesheim, Institute of Computer Science
Samelsonplatz 1, 31141 Hildesheim, Germany

²Kaiserslautern University
P.O. Box 3049, 67663 Kaiserslautern, Germany

³German Research Center for Artificial Intelligence
Trippstadter Strasse 122, 67663 Kaiserslautern, Germany
{firstname.lastname}@dfki.de

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

Abstract To support the early conceptualization phase in architecture with computer-aided solutions, in particular, with retrieval systems that can find similarly structured building designs in comprehensive collections of such designs, a number of approaches were presented to date. In the Metis project two retrieval coordination approaches (coordinators) were developed to govern the search of similar (sub-)structures of architectural designs. The main task of both coordinators is to select the retrieval method that is appropriate for the given user query. First approach is a standalone service that uses rules only to coordinate the retrieval and can use subgraph matching and database search methods, whereas the second one is rule- and case-based and is part of a distributed system for case-based retrieval of architectural designs. We compared both coordinators in a user study to find out which strengths and weaknesses both coordinators currently possess, and for which retrieval scenarios of the architectural conceptualization phase they could be appropriate. The results showed that the complexity of the particular scenario and the purpose of search are the main points that differentiate both coordinators. The rule-based coordinator performed better when a search for exact (sub-)structures was required, whereas the rule- and case-based coordinator is appropriate for queries aimed to be used for exploration and general search for inspiration. Visualization of the results of both coordinators is in need of improvement.

Keywords: CBR applications, CBR and creativity, case-based agents, case-based design, case-based retrieval, case-based coordination

1 Introduction

The early design phase in architecture is characterized by analysis of design ideas, that architects iteratively elaborate by themselves, and references found in corresponding specialist material. The traditionally established and nowadays still widely used conceptualization approach is a pen-and-paper-based design phase with iterative comparison of the progress with the references in the printed media. By comparing similar design ideas a design can be evaluated, used as inspiration or explicit design solution regarding different criteria. The computer-aided retrieval of similar design ideas in digital collections of such designs can be a significant improvement for the early conceptualization phase of architecture. It can help an architect to speed up the design process by immense reduction of time spent for search and so make it more efficient and productive. In cases, where the currently used retrieval system implements multiple search algorithms, a coordination approach is needed, that selects the proper algorithm and/or strategy based on the (user-generated) data contained in the search request.

In this work we compare two different retrieval coordination approaches, both developed, among other services, for the *Metis* project (see Section 2 for more information about the project). The first one is the rule-based only coordination, which selects a suitable retrieval method based on the implemented rulesets. This coordinator was developed by the KSD¹ research group of the Technical University of Munich (*TUM*) and will be referenced as the *KSD Coordinator* in further sections of this paper. The second is the case- and rule-based coordination agent of the distributed case-based retrieval system *MetisCBR* [3], developed by the German Research Center for Artificial Intelligence (*Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI*). This coordinator is a case-based agent (*CBA*) and is the main node of the retrieval process within the system. It will be referenced as the *MetisCBR Coordinator* in further sections of this paper.

This paper is structured as follows: first the related external and internal work of the *Metis* project will be presented. After that, in Section 3, both retrieval coordination approaches will be described in detail, their main features and abilities will be presented.

In the Section 4 we provide a comprehensive cross-evaluation of both coordination techniques. By means of applying a number of queries created during the study we will take into account the computed similarity values of the building designs retrieved by both coordinators, the subjective opinion on quality of the result set according to the architectural informatics experts, the overall number of results, and other details. The main purposes of this pilot evaluation is to find out which strengths and weaknesses both coordinators currently possess (in order to coordinate their development in the future) and to determine which coordination technique is currently the most suitable for which user scenario.

The conclusion and future work sections close this paper and give an overview of the presented study and a short description of planned work in the *Metis* project. This study is also part of the PhD work of the first author of the paper.

¹ Knowledge-Supported Design

2 Related Work

To find essential solutions in order to provide the computer-aided support of the early conceptualization phase in architecture, an interdisciplinary basic research project *Metis – Knowledge-based search and query methods for the development of semantic information models for use in early design phases*² was initiated by the DFKI and the TUM³. The project unites following main research areas: case-based reasoning (*CBR*), multi-agent systems (*MAS*), computer-aided architectural design (*CAAD*), and building information modelling (*BIM*).

The work, which has been accomplished in the project to date, consists of implementation of different search techniques, query builder interfaces, and supporting services, such as databases. Besides of that, theoretical research has been conducted in the project as well. The two important theoretical approaches that were created during the project-related research are the *Semantic Fingerprint* [13] paradigm and *AGraphML* [12] (see Section 3). As databases, among others, the *Neo4j* graph database with building design graphs, the content management system (CMS) *mediaTUM* for graphical representations of the designs, and the *Open Source BIM Server* can be named. The implemented query builder interfaces include the web-based floor plan editor (*Metis WebUI*) [4], a touchable application, and iPad and Android applications. The currently implemented search techniques include the subgraph matching algorithms together with the case-based retrieval techniques [2] of *MetisCBR* and an index-based retrieval method with the *Cypher* language queries of *Neo4j*. The subgraph isomorphism techniques include the implementations of the VF2 approach [7] and of the enhancement [20] of the original Messmer-Bunke algorithm [15], implemented under the name *GML Matcher*.

Prior to *Metis*, much essential work has been done in the domain of support of the early phase of architectural design. These projects and research initiatives left a legacy of methodologies and applications that we could build on. In [9] and [17] overviews of the applications are provided. An essential work of Richter [16] enhances this research by providing detailed in-depth descriptions and analysis of the approaches in particular and of *CBR* in architecture in general. Noticeable approaches are *FABEL* [19], *CaseBook* [10], *DIM* [11] or *CBArch* [6] inter alia. For case representation, *VAT* (Visual Architectural Topology) [14] provides a semantic way of representation, based on ontological expressions of floor plan topologies.

For the *MAS* area, the work of Anumba et al. [1] is one of the essential publications that provides insights into embedding of *MAS* in construction, architecture, and related domains. Application examples, as well as theoretical foundations, are presented and described in detail.

² *Metis – Wissensbasierte Such- und Abfragemethoden für die Erschließung von Informationen in semantischen Modellen für die Recherche in frühen Entwurfsphasen.*

³ The project is funded by the German Research Foundation (*Deutsche Forschungsgemeinschaft, DFG*).

3 Examined coordinators and Their Features

In this section we present the features of the coordinators we are evaluating in this work, using the categories of *retrieval-related aspects* (such as *fingerprints*, *weighting*, and *query protocol*) and *additional aspects* (such as *extensibility*). First, we provide an overview over foundations that are common for both coordinators.

- *Building Information Modeling (BIM)* is an approach for machine-interpretable modeling of buildings for the purpose of storing building information across the lifecycle of a building. BIM’s object-oriented concept is based on *parametric* objects that contain information about attributive geometric data of a building. A comprehensive information source about BIM is a handbook [8].
- *Semantic Fingerprint* [13] is a paradigm that describes a pattern structure for flexible, hierarchical and index-based definition of building design metadata. Semantic Fingerprint is related to BIM and is intended to extend BIM for use in modern computer-aided architectural applications. The fingerprint patterns can be used for description as well as for querying or comparison of building designs in such an application. The characteristics of a fingerprint (*FP*) can use semantic information of a building, topology, or relations (direct or adjacent) between rooms. Commonly, FPs can be represented as labeled floor plan graphs with rooms as nodes, and room connections as edges. For the Metis project a list of implementable FPs was defined. The Table 1 provides an overview over the FPs currently available for *both* coordinators.
- *AGraphML* is the extension of GraphML [5] that has an architectural specification [12] as its underlying structure. AGraphML was developed by TUM and DFKI for the projects of the KSD research group (like the Metis project). It is used for the XML-based representation of semantic fingerprint-based floor plan graphs (e.g., as part of query format, see also Sections 3.1 and 3.2).

Fingerprint	Description	Specifics
Room Count	Number of rooms	No connections between rooms and no labels specified
Edge Count	Number of edges	No room information specified
Room Graph	Anonymous representation of rooms and edges	No labels specified for rooms and edges
Room Functions	Labels of rooms	No connections between rooms, room labels are specified
Room Semantics	Emphasis on room semantics	Rooms information is complete, no edge labels specified
Passages Semantics	Emphasis on edge semantics	Edges information is complete, no room labels or names specified
Semantic Connections	Complete graph	All information about rooms and edges is available

Table 1: Fingerprint patterns currently available for both coordinators.

3.1 KSD Coordinator

The KSD Coordinator Web Service [18] provides a comprehensive *middleware* application for search requests from compatible front-end clients to different retrieval approaches in order to find similarly structured building designs. The key features of the KSD Coordinator are the specific *language for query construction* (with support of search constraints and similarity assessment definitions) and the underlying *rule engine* that is aimed to determine which retrieval method is the most suitable for the given query. The KSD Coordinator also provides an own user interface with configuration and query playground among other things. The next sections provide an overview over the KSD Coordinator’s functionalities.

Retrieval-Related Aspects The KSD Coordinator is able to trigger a number of search techniques, where for each of these techniques a special particular thread/process, an *agent*, is responsible. The selection of agents depends on the fingerprints determined in the query. For each fingerprint an assigned agent type exists. Currently three of these types are available:

- *Mediatum Agent*: an agent that retrieves the database of the CMS mediaTUM directly and is suitable for FPs where a floor plan metadata attribute is the main search criterion (**Room Count** and **Edge Count** FPs).
- *GML Matcher*: an agent that uses the extended Messmer-Bunke algorithm [20] for retrieval of isomorphic subgraphs using the complete graph information. This agent is used for the **Semantic Connections** FP.
- *Neo4J Agent*: a flexible agent for retrieval of similar graphs, where information can be incomplete. Queries the Neo4j database directly with *Cypher* queries and is used for **Room Graph**, **Room Functions**, **Room Semantics**, and **Passages Semantics** FPs.

Retrieval results are weighted during the *Ranking* process of the KSD Coordinator. The final rank of a result is computed by the sum of all conditional ranks of this result. A conditional rank is a product between an indicator function and the sum of a fixed value and the product of the weight and similarity value.

The query protocol (or *query language*) of the KSD Coordinator is XML-based and defines a pipeline of components that will be decomposed and simplified during the retrieval process. Three types of such components (blocks) exist:

- **let**: Provides AGraphML-based graph definitions that can be declared by a variable. Optional block.
- **select**: Defines data contained in the result set. Optional block.
- **where**: A mandatory block. Defines which retrieval *conditions* should be applied to the query. Represented as a Boolean expression in XML format. A condition can contain a fingerprint reference (if a graph was provided in the **let** block), a defined retrieval method, or a specific metadata attribute.

The conditions from the **where** block will be later decomposed and parsed by the *rule engine* of the KSD Coordinator. The assessment of the conditions depends on the *rule scripts* contained in the rule engine. The rule scripts define which search agents should be triggered for a given condition.

Additional Aspects The KSD Coordinator can be extended with new agents and new fingerprints, inter alia. A new agent definition can be added by providing a corresponding agent configuration, agent lifetime class, and the data source class. For addition of a new fingerprint, a definition of a corresponding graph equivalence concept and an associated agent are required.

A feature of *caching* is available for the KSD Coordinator that allows for saving intermediate results of previous queries. A set of caching rules includes the caching and reuse options for *full query caching* and *data source caching*. The *timeout* feature allows for setting a maximum query execution time for an agent.

3.2 MetisCBR Coordinator

In MetisCBR, the general task of retrieval coordination is distributed among the *Coordinator* agent and its helper agents *SubCoordinator* and *Timeout*. The two helpers were created to reduce the overload of work of the coordination agent. Given this context, we can also speak of the *coordination team* of MetisCBR.

The actual coordination in MetisCBR is divided into two steps: *rule-based* and *case-based*. In the first, rule-based step, the ruleset of the MetisCBR Coordinator determines, based on the *user-generated* data from the query, which FPs will be used for the current retrieval. Then, in the case-based step the myCBR-based mechanism of the MetisCBR Coordinator tries to find the most similar QUERY case instance in its case base. If the similarity value is sufficient, the results achieved by this previously saved request, will be presented to the user, without starting a new retrieval process. After the evaluation of the results, if the user prefers to conduct a new retrieval anyway, the retrieval will take place. The results (old/saved in the similar case, or achieved by a new search) will be added to the new QUERY case instance and saved in the coordinator's case base. If the user has not determined which FPs should be applied during the search, but the most similar QUERY case instance had FPs applied, then the user will be informed about this fact during the output of the results. The task of retrieval of the most similar QUERY and saving the results is optional and can be disabled in the configuration of the system. It can also be seen as the *caching* or *indexing* process. The complete procedure of query processing is shown in Figure 1.

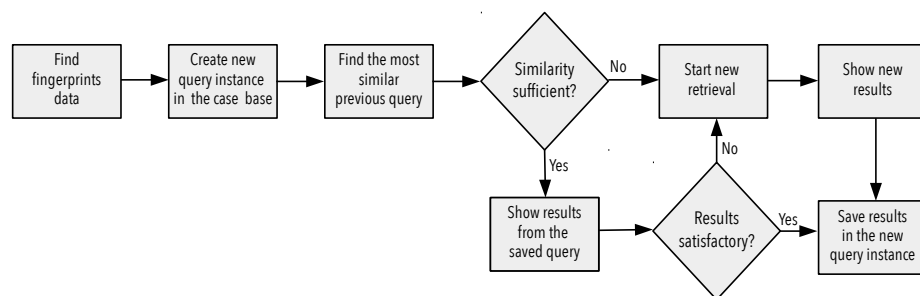


Figure 1: Query processing steps of the MetisCBR Coordinator.

Retrieval-Related Aspects The MetisCBR Coordinator has access to different retrieval methods implemented in the MetisCBR core. The methods can be classified into two main classes: *generic* and *fingerprint-related*. Both of these classes use the same set of retrieval strategies, the main difference between the classes is the purpose of use: the category of generic methods is used if no FPs were applied to the query, whereas the fingerprint-related methods are obviously applied if FP information is available in the query. Another, more practical, difference is the set of attributes used for the comparison during the search: for non-fingerprint methods all of the attributes are used, for each fingerprint method a set of suitable attributes is used to find the most similar cases. The underlying CBR domain model [2] defines the structure of the cases (building designs). The currently available retrieval strategies include two following types:

- The multi-step *Basic strategy* (described in [2] as well) for fingerprints considered complex (**Room Graph**, **Room Semantics**, **Passages Semantics**, or **Semantic Connections**) and deep search for queries without fingerprints.
- The single-step *metadata strategy* that uses the floor plan metadata information only and is applied for fingerprints considered simple (**Room Count**, **Edge Count**, **Room Functions**) and for the fast search without fingerprints.

The weighting of fingerprints is applied according to the weights set by the user during the query building process. The weights should sum to 1 and will be multiplied with each similarity value of the result set of the corresponding fingerprint query. The weights assignment influences positioning of results in the final result set, where result sets of all of the fingerprints are combined.

The query format of the MetisCBR Coordinator makes use of the protocol developed for query construction in the MetisWebUI. In this protocol the AGraphML representation of the query is wrapped by the request tags that include the information about FPs as their child elements as well (see Listing 1).

```
<?xml version="1.0" encoding="UTF-8"?>
<searchrequest>
  <agraphml>
    <graphml>...</graphml>
  </agraphml>
  <fingerprint name="Room_Count" weight="0.4" />
  <fingerprint name="Passages" weight="0.6" />
</searchrequest>
```

Listing 1.1: Query protocol of MetisCBR (fingerprints are exemplary).

Additional Aspects Currently, the MetisCBR coordination team possesses the *open-box* extensibility feature. New behaviours and functions can be added and later changed inside the source code of the agent. Also, the addition of new helper agents is possible: either by extension of the source code of MetisCBR or by communication with the coordinator from other compatible agent platforms.

Also, a *timeout* feature is available for the MetisCBR Coordinator. The helper agent Timeout checks periodically the activity of the retrieval processes (containers), i.e., if it has finished its task within the defined amount of time.

4 Comparative Evaluation

To analyse the current state of both coordinators we conducted a comparative user study that consisted of creating different floor plan queries with the Metis WebUI and evaluating the results returned by each of the coordinators. To accomplish this task, the CAAD experts of the Metis project were asked to take part in the study as representatives of the architectural research area. For the study, a special setting was prepared: the previously mentioned Metis WebUI, and the input and output interfaces of both coordinators. The search options of both coordinators were set to equalize the retrieval abilities best possible:

- No caching or indexing should be used.
- The FPs **Room Count**, **Edge Count**, **Room Functions**, and **Semantic Room Connections** can be used (due to some technical problems of the KSD Coordinator other FPs could not be used).
- The weighting can be set for queries with multiple FPs.

The datasets of both coordinators included the building design graphs imported from the Neo4j database. Due to the technical restrictions and differences in the nature of import, the exact number varied for both coordinators, but can be estimated to have 200 as the approximate value for all FPs of the MetisCBR Coordinator and the **Semantic Connections** FP of the KSD Coordinator. For the FPs of the KSD Coordinator that were mapped to the Mediatum Agent ≈ 400 , and to the Neo4j Agent ≈ 500 building designs were used, as no import was needed, that is, the technical restrictions were also not applicable.

The actual process of the evaluation was divided into two phases. In the first phase a *storyline query* was created: during an iterative multistep process the initial / previous query was modified according to the view of the architect, the inspiration from the results, and requirements of this type of floor plan, and then used in the following step. Each of the iterations was related to the initial purpose (i.e., *scenario*) of the query. In the second phase separate *single queries*, which had no relation to each other, were used, each of these single queries had its own scenario. For each of the queries of both query phases the participants were asked to fill a questionnaire to review, estimate, or rank the overall results and the general behaviour of each of the examined coordinators, in order to provide their subjective opinion on the outcomes of the respective retrieval scenario. Following questions were included in the questionnaire:

- In general, are the results appropriate for this query? [scale 1(no)-5(yes)]
- Is an improvement of the results for this query required? [yes/no]
- Is it possible to mark one of the results as the *best* one? [yes/no]
- Can one of the results be used as a *template* for the next query? [yes/no, applicable only for the storyline query]
- Do the results contain a certain pattern or model? [yes/no]

For each of the answers it was also possible to leave an additional comment to provide an explanation of the opinion.

Besides of the subjective analysis, the *computed similarity values* (more precisely, the average similarity of all returned results and of the first 10 results), the *total number of results*, and the *inclusion of results of all selected FPs* in the returned result sets were also part of the evaluation.

The results of the retrieval were presented with the visualization method currently implemented in the corresponding coordinator interface (see Figure 2): the *graph representation* with information about nodes and edges in the KSD Coordinator, and the *pre-rendered graph* (with information about nodes only) along with the separate *graphical representation* in the MetisCBR Coordinator.

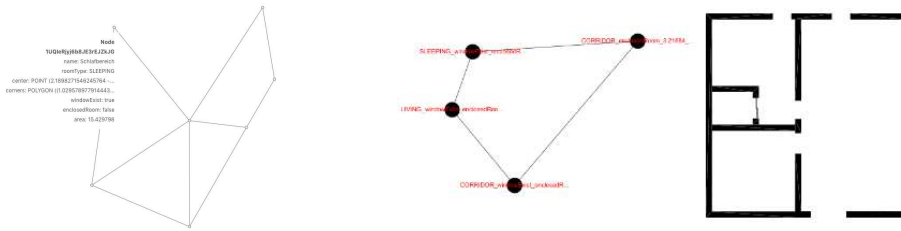


Figure 2: The result visualization methods implemented in the coordinators at the time of the study. On the left side the graph representation of the KSD Coordinator with information about a node. On the right side the pre-rendered graph and the separate graphical representation of the MetisCBR Coordinator.

4.1 Queries and Results

The results of the study will be presented divided into two parts, according to the number of query phases. We compare the outcomes of the questionnaire and the computed values by providing a summary for each of these result categories.

Storyline Query For the storyline query (see Figure 3) a scenario of an *apartment for elderly married couple* was selected by participants. For this query the participants decided to make two search requests in each iteration: first the request with the **Semantic Connections FP** and then the addition of the **Room Functions FP** in the next request (except the first iteration).

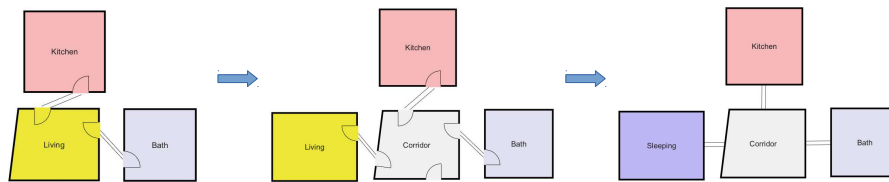


Figure 3: The iterative scenario of an apartment for elderly married couple. Light red is used for the **KITCHEN** room type, yellow for the **LIVING** room, grey for **CORRIDOR**, light violet for **BATH**, and dark violet for the **SLEEPING** room.

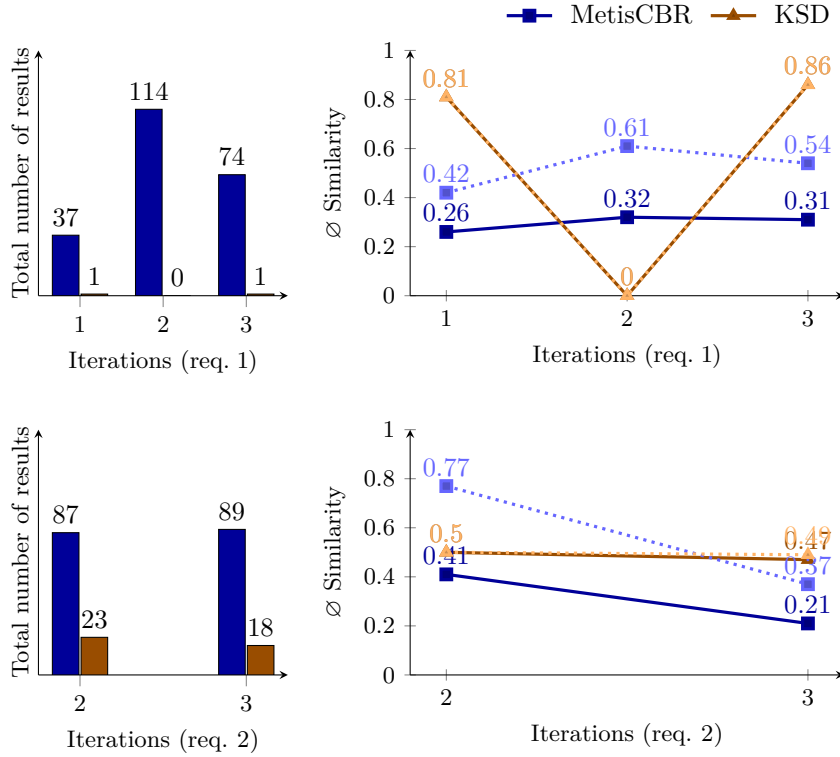


Figure 4: The results of the computed values of *both* search requests of the *storyline query*. In the first row the results of the first request (**Semantic Connections FP**), and in the second row of the second request (**Semantic Connections and Room Functions FPs**), are presented. The dotted lines represent the average similarity of the *first 10* results. In the third iteration the participants applied weighting: 0.3 for the **Room Functions FP** and 0.7 for the **Semantic Connections FP**.

Query and Coordinator	Results appropriate? 1(no)-5(yes)	Results improvement required?	Best result available?	One of the results is suitable as template?	Pattern in results
<i>Iteration 1</i>					
MetisCBR	2	yes	yes	yes	yes
KSD	5	yes	yes	yes	no
<i>Iteration 2</i>					
MetisCBR	2	yes	no	no	no
KSD	2	yes	yes (req. 2)	no	yes
<i>Iteration 3</i>					
MetisCBR	2	yes	no	no	no
KSD	2.5	yes	yes	no	yes

Table 2: Results of the questionnaire for the storyline query. Rank or estimation is accumulated or averaged if result sets for both requests were not empty.

In general, the results of the storyline query (see Figure 4 and the Table 2) confirmed the evident assumption that the exact matching approach of the GMLMatcher (Messmer-Bunke-Algorithm) would find the exact (sub-)structures among the graphs and can answer the question if such a structure is available in the database at all, so that the user can take a look how it is implemented in another floor plan. In contrast to this, the MetisCBR Coordinator returned quantitatively more results in each iteration, which provided much more space for exploration of similar designs. By adding the **Room Functions** FP in second and third iteration the KSD Coordinator increased the number of results and provided a sufficient number of explorable designs, where the room setting of the query and of the result are equal. Noticeable is also the fact that the average similarity of the first 10 results of the MetisCBR Coordinator is noticeably higher than this value of its all results, whereas for the KSD Coordinator the value remains (almost) equal in both measurements.

The questionnaire answers and corresponding comments and explanations confirmed that the number of results for the KSD Coordinator is currently an issue when using the **Semantic Room Connections** FP only. For example, this fact had the biggest influence on the question if the improvement of results is required. A similar problem also exists for the MetisCBR Coordinator: it has returned too much results and the most useful ones were not high positioned in the overall result set, so that a longer exploration was needed to find them. Regarding the question if a result can be considered a suitable template for the next iteration, both coordinators were equal, for both coordinators the separated / not graph-to-floor-plan mapped visualization was a problem in this particular case. The patterns were recognized in all iterations, but not for each coordinator: in the first iteration a pattern of replacement of the **BATH** room type by **CORRIDOR** was noticed in the result set of the MetisCBR Coordinator, whereas no such patterns were seen in other iterations. In contrast to this, the result sets of the second and third iteration of the KSD Coordinator contained patterns, in particular, a pattern of greater number of rooms was recognized.

Single Queries For the *single queries* part, three different scenarios were created by participants (see also Figure 5). For this query type, the participants also decided to make two search requests for each of the queries: first the request with the **Semantic Connections** FP and then the addition of the **Room Functions** FP in the next request.

- *Query 1: Bungalow* – A single-storey house with the working room placed at a slight distance to other rooms.
- *Query 2: Connection LIVING-SLEEPING* – The structure of living and sleeping room connected through a **DOOR** within other floor plan structures should be found to explore different implementations of this connection.
- *Query 3: Student Dormitory* – The building entry situation with three corridors connected through doors should be found. The architect is certain that a building design with such situation exists in the graph database.

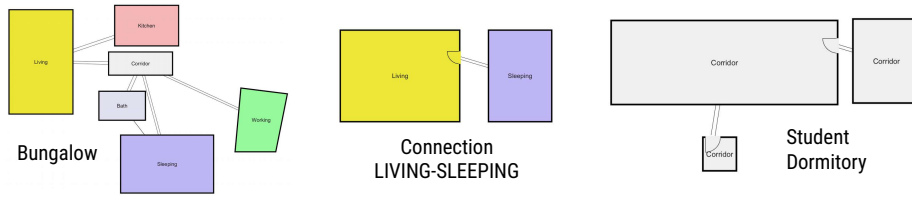


Figure 5: Single queries. Light red is used for the KITCHEN room type, yellow for the LIVING room, grey for CORRIDOR, light violet for BATH, dark violet for SLEEPING, and green for the WORKING room.

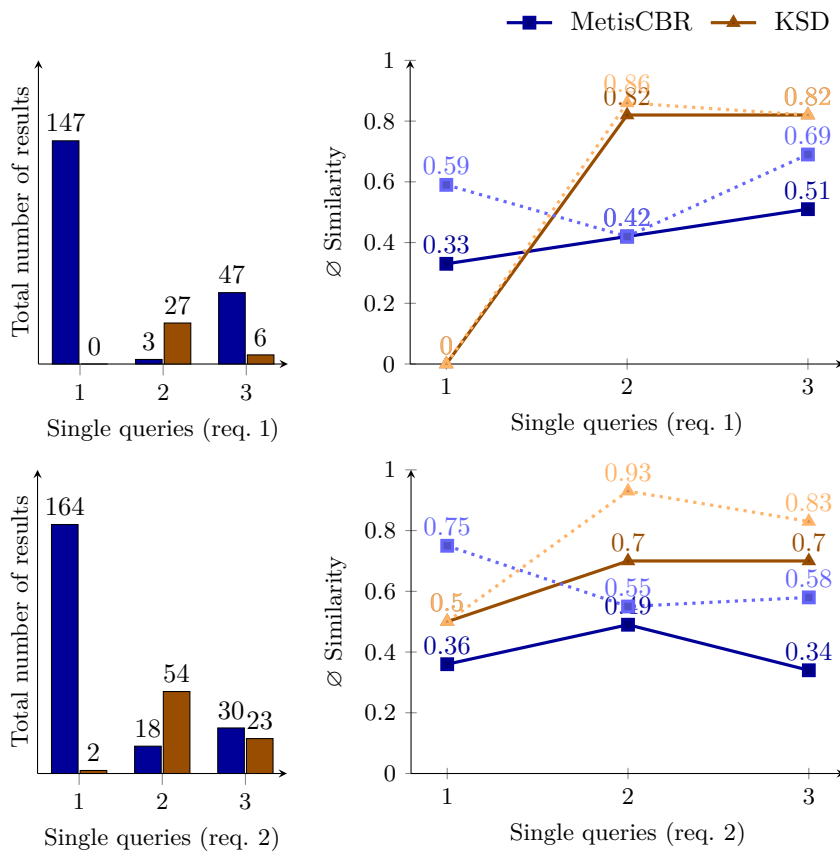


Figure 6: The results of the computed values of *both* search requests for each of the single queries. In the first row the results of the first request (**Semantic Connections FP**), and in the second row of the second request (**Semantic Connections and Room Functions FPs**), are presented. The dotted lines represent the average similarity of the *first 10* results. Note the fact, that in the third query (*Student Dormitory*) the participants decided to apply weighting: 0.8 for the Room Functions FP and 0.2 for the Semantic Connections FP.

Query and Coordinator	Results appropriate? 1(no)-5(yes)	Results improvement required?	Best result available?	Pattern in results
<i>Query 1: Bungalow</i>				
MetisCBR	2	yes	no	yes
KSD	2.5	yes	no	no
<i>Query 2: Connection LIVING-SLEEPING</i>				
MetisCBR	2	yes	no	no
KSD	4.5	yes	no	yes
<i>Query 3: Student Dormitory</i>				
MetisCBR	2	yes	no	yes
KSD	4	yes	yes	yes

Table 3: Results of the questionnaire for the single queries. Rank or estimation is accumulated or averaged if result sets for both requests were not empty.

The evaluation of results of the single queries has shown that the advantages of the CBR-based retrieval for similar architectural designs are noticeable when it comes to queries with more complex structures. For example, for the *bungalow* query, the MetisCBR Coordinator returned much more results than the KSD Coordinator. Also, for this query, the MetisCBR Coordinator was able to return results for each FP, whereas the KSD Coordinator could only find results for the **Room Functions** FP (when it was added). In contrast to this, in the second query, which consisted of the very simple connection *LIVING-SLEEPING*, the KSD Coordinator could find more results for exactly this connection, using its advantages of the subgraph matching. The same applies to the *student dormitory* query, where the KSD Coordinator was able to find exactly the floor plan concept the architect was looking for. Though, the MetisCBR Coordinator has found more similar cases, the exact match was not among them.

The subjective opinion of the architects, inferred from the questionnaires for single queries, showed that the MetisCBR Coordinator returned some satisfactory results for the bungalow query, only when the **Room Functions** FP was applied. Nevertheless, the question if there is a *working room* that is placed at a slight distance to other rooms could not be answered exactly. The same is applicable for the results from the KSD Coordinator, where one of the two results of the **Room Functions** FP was considered inspirational. For the *student dormitory* query, the usefulness of the results of the KSD Coordinator achieved by exact matching was noticeable. As mentioned above, the MetisCBR Coordinator was not able to find the exact concept, though some results could be considered very similar as they consisted mostly of the room setting given in the query (but not as part of a very complex floor plan). The visualization problem was noticed for the single queries as well as for the storyline queries. For example, the results of the *LIVING-SLEEPING* query could be evaluated more thoroughly, but the missing of concrete edge information in the visualized MetisCBR results and missing representations of the actual floor plan for the KSD Coordinator results could not provide the needed information, so that no result could be considered best in the result sets for this query of both coordinators.

5 Conclusion and Future Work

In this paper we presented two retrieval coordination approaches: the rule-based KSD Coordinator and the rule- and case-based MetisCBR Coordinator. Both of the coordinators are integrated in the infrastructure of the KSD research group and able to access different retrieval methods for search for architectural designs with similar structures. Both coordinators use agents or related techniques for delegating of the actual retrieval tasks, the retrieval strategy depends on the user-defined architectural semantic fingerprint data in the query. AGraphML is used by both coordinators as the format for graph representation in the query.

We evaluated both coordinators in a comparative study that included two separate query types: an iterative storyline query where each iteration is related to a common scenario and is based on data from the previous iteration, and single queries where each of them has its own scenario. The results of the evaluation showed that the situations where an exact match of the structure is needed are more suitable for the KSD Coordinator as it can trigger the subgraph matching method based on the original Messmer-Bunke algorithm, whereas the MetisCBR Coordinator can provide more space for exploration of the similar floor plans contained in the data-/case base. For both coordinators, a problem of visualization exists that has not allowed for more thorough evaluation of results in some cases. Our assumption is, that the combination of both coordinators, where each of them is responsible for special fingerprints, would currently provide the most comfortable way to support the early conceptualization phase in architecture using the techniques developed for the Metis project.

In future studies we are going to evaluate the implemented retrieval methods in a more comprehensive way taking more search techniques into account (including more exact and inexact graph matching methods). Also, a number of different result visualization methods will be further developed and evaluated.

References

1. Anumba, C., Ren, Z., Ugwu, O.: Agents and multi-agent systems in construction. Routledge (2007)
2. Ayzenshtadt, V., Langenhan, C., Bukhari, S.S., Althoff, K.D., Petzold, F., Dengel, A.: Distributed domain model for the case-based retrieval of architectural building designs. In: Petridis, M., Roth-Berghofer, T., Wiratunga, N. (eds.) Proceedings of the 20th UK Workshop on Case-Based Reasoning. UK Workshop on Case-Based Reasoning (UKCBR-2015), located at SGAI International Conference on Artificial Intelligence, December 15-17, Cambridge, United Kingdom. School of Computing, Engineering and Mathematics, University of Brighton, UK (2015)
3. Ayzenshtadt, V., Langenhan, C., Bukhari, S.S., Althoff, K.D., Petzold, F., Dengel, A.: Thinking with containers: A multi-agent retrieval approach for the case-based semantic search of architectural designs. In: Filipe, J., van den Herik, J. (eds.) Proceedings of the 8th International Conference on Agents and Artificial Intelligence. International Conference on Agents and Artificial Intelligence (ICAART-2016), February 24-26, Rome, Italy. SCITEPRESS (2016)

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. Cordella, L.P., Foggia, P., Sansone, C., Vento, M.: A (sub) graph isomorphism algorithm for matching large graphs. Pattern Analysis and Machine Intelligence, IEEE Transactions on 26(10), 1367–1372 (2004)
8. Eastman, C., Eastman, C.M., Teicholz, P., Sacks, R.: BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors. John Wiley & Sons (2011)
9. Heylighen, A., Neuckermans, H.: A case base of case-based design tools for architecture. Computer-Aided Design 33(14), 1111–1122 (2001)
10. Inanc, B.S.: Casebook. an information retrieval system for housing floor plans. In: The Proceedings of 5th Conference on Computer Aided Architectural Design Research (CAADRRIA). pp. 389–398 (2000)
11. Lai, I.C.: Dynamic idea maps: a framework for linking ideas with cases during brainstorming. International journal of architectural computing 3(4), 429–447 (2005)
12. Langenhan, C.: A federated information system for the support of topological bim-based approaches. Forum Bauinformatik Aachen (2015)
13. 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)
14. Lin, C.J.: Visual architectural topology. In: Open Systems: Proceedings of the 18th International Conference on Computer-Aided Architectural Design Research in Asia. pp. 3–12 (2013)
15. Messmer, B.T., Bunke, H.: A decision tree approach to graph and subgraph isomorphism detection. Pattern recognition 32(12), 1979–1998 (1999)
16. Richter, K.: Augmenting Designers' Memory: Case-based Reasoning in Architecture. Logos-Verlag (2011)
17. 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)
18. Roith, J., Langenhan, C., Petzold, F.: Supporting the building design process with graph-based methods using centrally coordinated federated databases. In: 16th International Conference on Computing in Civil and Building Engineering (2016)
19. Voss, A.: Case design specialists in FABEL. Issues and Applications of Case-based Reasoning in Design pp. 301–335 (1997)
20. Weber, M., Langenhan, C., Roth-Berghofer, T., Liwicki, M., Dengel, A., Petzold, F.: Fast subgraph isomorphism detection for graph-based retrieval. In: Case-Based Reasoning Research and Development, pp. 319–333. Springer (2011)