EXPLORING PERFORMANCE QUERY SPACE

Smita Gupta[1] and Ardeshir Mahdavi[2]
[1]School of Architecture, Carnegie Mellon University, Pittsburgh, PA
[2]Department of Building Physics, Vienna University of Technology, Austria

ABSTRACT

This paper proposes a perspective to view and structure the performance queries in terms of a multi-dimensional query space. The classification of the queries is likely to render them more suitable for analysis, resulting in enhanced responses through selection and execution of appropriate computation tools and techniques. Performance queries can be broadly classified into three types; namely a) those involving the determination of the values of specific performance indicators (e.g. energy use and thermal comfort); b) those addressing cost implications of alternative design options; c) those pertaining to compliance with (local, national, or international) codes and standards. The paper demonstrates the way in which a performance query space can serve as the entry point for a network-based multi-disciplinary performance-based design evaluation environment.

INTRODUCTION

Research in the performance simulation field in the past has been motivated by the facts that: a) performance feedback can improve a design; b) simulation tools can provide the efficient feedback; c) designers typically do not use such computational tools. In effect most effort has gone into the development of suitable tools, enhancing their environment and their user interface to address these problems, thereby affording a "tool-centric" view to performance simulation research. An alternative yet seldom explored view point starts with the user perspective and poses questions about the typical queries that are involved in performance assessment.

In a typical design process the architect has questions about the building's performance and would usually rely on his/her own judgment/calculations or approach consultants, who would be able to perform these and convey a suitable answer, whether through experience or rendering the services of simulation engines. These queries range into a variety of performance domains, energy, acoustic, lighting and more, which the consultants have to assemble in order to derive the answer.

The main question has eluded the researchers and developers in the field, since most of the effort has been concentrated on making better tools. In the past the problems related to most of the stages of performance analysis have been addressed in one form or another. The detailed and comprehensive tools of course have been a long time focus and there exists a solid base of knowledge and research that has developed and refined over time the various algorithms (such as TRNSYS, BLAST). Even the relatively simplified and quick use tools have been well developed and possess, in part, quite intuitive interfaces. On another level the question of integrated building performance over various domains has taken the notch up and provided for a higher level of functionality (COMBINE, ESP-r, EnergyPlus). Furthermore the case for integrated yet distributed performance computing has been well addressed and demonstrated as a prototype by the SEMPER team (Mahdavi et al. 2002). The IAI's IFC efforts which are underway aide the cause for interoperability of the semantic information required for the performance computing. Another level up is the DAI (Augenbroe et al, 2003) effort, which sought to address the analysis scenarios and present analysis functions as a basis to interface the various tools of computation through a process-centric approach. But taking a couple of steps back to get a broader view of the picture, there seems to be the missing piece of the intentionality of the whole development approach. The question that emerges is: What do the architects/designers want to know regarding the performance queries of their building design and how best to address their query.

PERFORMANCE QUERIES

The performance of a building with regards to the energy and other domains is often an unanswered question in the minds of architects. The inclusion of experts in the early design stages is often not sought. But the benefits of early performance feedback are irrefutable in the energy end-use of the building. So it becomes pertinent to facilitate the addressing of these queries. Evidence suggests that the performance queries typically spread over a broad range in both content and intent. More over these queries are based in the natural language rather than in the specific jargon of performance domain. Some examples of queries can be seen in figure 1. These were collected through an informal survey of architects and energy engineers/analysts

.

- What is the energy consumption of the building?
- Is there enough daylighting?
- What is the best orientation of the building?
- What is the energy consumption of a particular building?
- Compare between the two given massing strategies
- What shading strategy to use – blinds, overhangs, fins, dynamic louvers etc.
- What percentage of glazing is optimum
- Does the building comply with Title 24?
- What is the pay back period of using ground source heat pump?
- Which system is better (from a list of systems)
- What is the LEED rating of the building?
- Best operating strategy (from a list of control strategies)
- What size of system to use?
- What is the maximum-minimum temperature in the space
- Is there adequate cross-ventilation?
- How much thermal mass is required for load shifting?
- How can I make this building's energy consumption 1/3rd of the typical building energy consumption?
- What is the performance of this new material with regard to energy consumption?

Figure 1 Some typical queries

On closer examination, the above queries can be broadly classified into three types; a) directly performance indicator based; b) cost related; c) and related to code compliance (for local authorities or predefined industry standards). This broad categorization helps in the resolution of the query with a view to compute the answers. This act of categorizing and simplifying the queries into their constituents, gives rise to the notion of a "query space" to locate them.

QUERY SPACE

The query space is proposed to have dimensions that help define the query and narrow it down to the level of simple computable entities. The basic five dimensions that help define a query are suggested as: a) the relevant performance domain, b) mode c) design information resolution d) extension e) context.

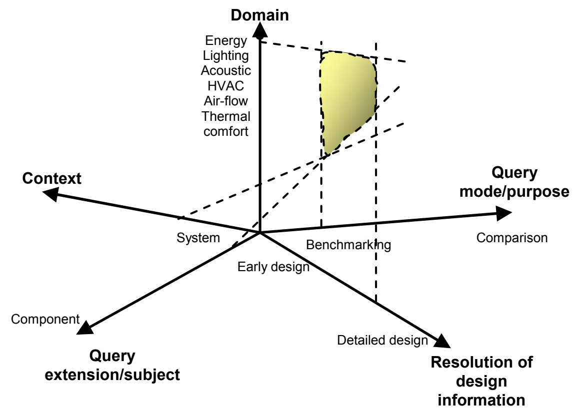

Figure 2 Query Space

The domains dimension classifies the query as addressing one or more performance domain, such as energy, lighting, air-flow, acoustic, thermal comfort, HVAC and others of the category. This helps identify the set of performance indicators that will need to be computed for the query.

The mode signifies whether the query is a benchmarking or a comparison. Benchmarking involves in general finding the value of one or more performance indicators. This could be just to appraise the performance of a component or check for compliance to a code or standard. Comparison could, on the other hand, require a set of parametric runs to arrive at a result.

The design information resolution is the dimension that will help match the simulation tool to the performance indicators that need computing. For example, a schematic design may or may not led itself to detailed air flow analysis, where as a thermal analysis based on degree day method or using simplistic calculation might be possible. This dimension will also help to determine the need for automated information augmentation service if required.

The extension of the query denotes the subject of query with regard to the building under consideration. The subject of the query could be a single sub-component of the whole building or the entire building or the building system. This, in conjunction with the mode of the query, can further narrow the query space. Figure 2 graphically shows the possible mode and extension descriptions of the building project.

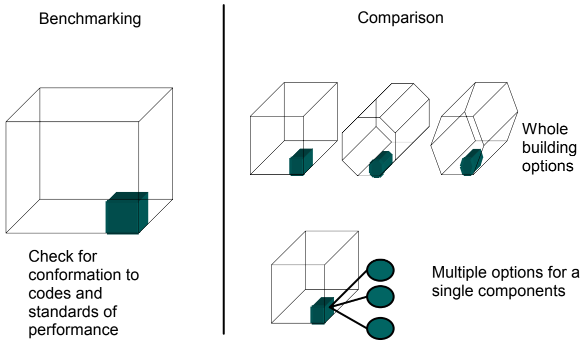

Figure 3 Query descriptions

The context of the query will place it geographically and globally in a location.

QUERY LANGUAGE

A major piece to the success of this query based approach will have to be the interpretation of the natural language query by the architect. This will need formalization in order to be recognized by the system. Performance Query Language (PQL) is a suggested approach that can aide the formalization of the syntax to map the requirements of a query. Drawing analogy from SQL (Structured Query language) for searching databases, PQL can serve as a means to search the query space and define the query in terms of the performance dimensions.

Perilinguistic studies in the computing world provide for a natural language based entry into a system, which can be analyzed and interpreted to carry out the needed actions. Various approaches have been explored in the arena of natural language recognition and parsing with respect to a domain specific knowledge base. The success of efforts such as clinical diagnostic tool - CLINAID (Kohout et al. 2001) and KANTOO - Multilingual Machine Translation System (Mitamura et al. 2002) can be used as examples to extend into the building performance domain.

The main idea is the presence of a keyword lexicon, specific to the knowledge domain which can be matched against the input query. The lexicon consists of all words recognized by the system, their grammatical categories, synonyms, and associations (if any) with domain objects. The final query can be one that is refined through an iterative approach over a keyword based search till it is well mapped into the query space. This methodology is also well developed and defined in the realm of web search engine and protocols. Once the query is iteratively mapped into the system and tied to specific performance indicators, the task is then reduced to that of identifying and gathering the service tools that can address it.

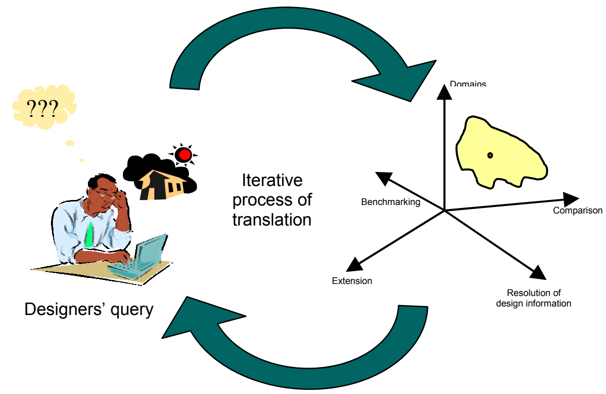

Figure 4 Iterative process of query refinement

Natural language recognition and parsing can break down the grammar of a sentence to derive the intention related to the performance domains. As an example, based on the few sample questions some keywords/phrases can be derived that convey the intent of the query (Table 1).

Table 1 Example of keyword/phrases

Keywords/phrases	**Intent**
Energy consumption	Calculate energy use
Daylighting	Calculate the daylight factor and/or illuminance
System size	Calculate the loads to size the system
Natural ventilation	Calculate rate of air change
Payback period	Calculate the monetary saving potential
Enclosure parameters	Calculate the attributes of thermal resistance
Shading	Calculate heating/cooling loads with and without shading
Glazing	Calculate the visual and/or thermal effect of a certain glazing type
Thermal mass	Calculate space temperature regimes
Space temperature	Calculate the spatial distribution of indoor temperature

The example in figure 5 illustrates the recognition of key phrases in a simple query and how it helps narrowing the query space. The designer can then be asked to verify additional information such as the design stage to determine the design resolution available and further refine the query point. This approach will help bring down the query to a set of performance indicators required and the information available to compute them. This set of information is valid enough to then progress further with the selection of service tools.

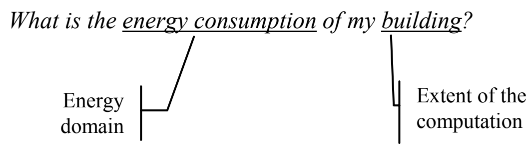

Figure 5 Example query interpretation

PERFORMANCE SIMULATION TOOLS

The various performance computation tools serve as service providers in this query centric process. They are identified with their capabilities and input requirements. Typically, the identity print of a service tool will declare its availability on the network along with its input requirements and the output performance indicators (as vectors) it can compute.

TOOL	INPUT REQUIREMENTS	OUTPUT
Name and location on the network	The building model resolution required along with weather data and other setup information	The actual performance indicators calculated as vectors.

Figure 6 Tool identity

Table 2 Example of tool identity prints

TOOL	INPUT REQUIREMENTS	OUTPUT
Name tag: Tool 1 @URL1	Schematic design building information (published data model) Weather conditions Number of people Type of activity	PMV PPD SET TSENS DISC
Tool 2 @URL2	Detailed design information Location of fenestration Artificial light source details	Daylight factor Illuminance
Tool 3 @URL3	Schematic or detailed design information Weather file Material properties	Hourly space temperatures Annual energy consumption

MAPPING

The process of refining the query will be iterative and based on the user inputs. Once the query is refined into the computable units in the form of performance indicators and all the other dimensions of the query space, it can be over-laid on a tool matrix to identify the required services. The tool matrix is essentially a tabulation of the tools and their computable performance indicators. These, when matched with the input requirements of available tools, give an opportunity to select or reject the services of the tool. In case the output requirements are met but the available inputs fall short, then additional service may be needed to augment the information. In certain cases, an automated augmentation service can add and generate any missing data that will be required for the computation.

Tool	Performance indicators					
	T_{sp}	E_{bldg}	RH	V_{sp}	n	n+1
Tool A	●	●				
Tool B		●	●			
Tool C			●	●		

Query 1: required computation for a mechanically conditioned bldg

Figure 7 The tool matrix overlap for simulating a naturally ventilated building

In figure 7 a query to compute the performance of a mechanically conditioned building is mapped on the tool matrix. This allows the system to select the three tools A, B and C to run the computation. The energy, hvac and air-flow domain as suggested by the performance indicators of the tools are required to comprehensively simulate the performance in this case. Another matrix that would decide the tool selection and suitability would be based on the input requirements for the tool to run the computation. If the requirements are not met the system could query and obtain an additional service to provide augmentation or in the lack of, defer the use of the tool.

CONCEPTUAL SYSTEM

Agent based networks provide the framework for such a system, where the service providers are all agents on the network and respond to calls for service when needed. Mutli-agent systems can work out action selection based on internal need (service request) and/or external stimulus (query initiation). The motif architecture (Liu 2001) presents such a system prototype, where the components cluster together in an architecture according to their data-exchange requirements.

The system to address this query based approach is envisioned as a networked society of agents that provide the various services. These services will range from the actual computation tools to the databases of required information, such as weather and material data. An additional service provider will be the mediator service, which is capable of coordinating the functioning and communication between the various other services. The role of the mediator service is similar to that of a broker to coordinate and collect the services of other agents.

The four basic types of agents for this society are: 1) the simulation tool 2) database service 3) mediator agents and 4) generative agents.

The simulation tools act as agents to provide the computation of performance indicators and are recognized by their identity print as described in the earlier section.

The database services are data providers to the system. They hold data such as weather information and material properties, which are requested any of the other agents in the society.

The mediator agents are the brokers of the society. They coordinate and assemble the services for the working of the system. In the proposed system the mediator agent will also serve to interpret the natural language query and generate the query space plot.

The generative agents automate any information augmentation as needed. For example, there can be an agent that generates the duct network for a given layout in the absence of one when detailed system simulation is requested by the query.

RETSINA as a potential framework offers an environment suitable for the implementation of this system (Sycara 2003). Its functional architecture consists of four basic agent types:

1. Interface agents - interact with users, receive user input, and display results.
2. Task agents - help users perform tasks, formulate problem-solving plans and carry out these plans by coordinating and exchanging information with other software agents.
3. Information agents - provide intelligent access to a heterogeneous collection of information sources.
4. Middle agents - help match agents that request services with agents that provide services.

Figure 8 shows an example of the internal communications in a conceptualized system with the query being external to the system.

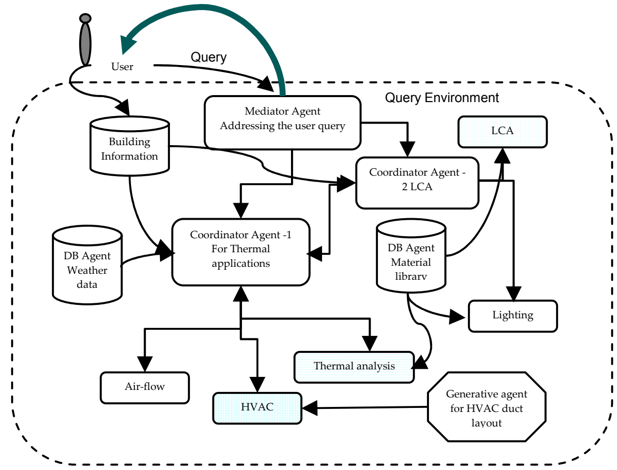

Figure 8 Sample of multi-agent performance services society

A flow through the query system can be exemplified by analyzing a query: what is the (predicted) energy use of a building?

The system receives the query and analyzes it through the various levels (Figure 9) of data that need to be ascertained in order to process the query. The initial natural language parsing PQL agent will derive the possible information to populate the various levels. The intent is to be able to derive the information at the lowest level, i.e. the last level under each category. Any pertinent information not apparent in the query sentence is extracted interactively from the user. This enables the system to suitably select the tool/s that will compute the results for the query. In this example, the domain is energy and the performance indicator to be computed is energy use. The mode and extent can be assumed to be benchmarking and the whole building respectively. The resolution of building information available is sought from the user, as well as the context. The simulation and database services providing the matching identity print will be selected to perform the computation. In this case the thermal energy analysis tool with the help from the weather database and the material properties database will be found suitable and will run the calculation for energy consumption.

Figure 10 illustrates a generic use-case information flow in the system.

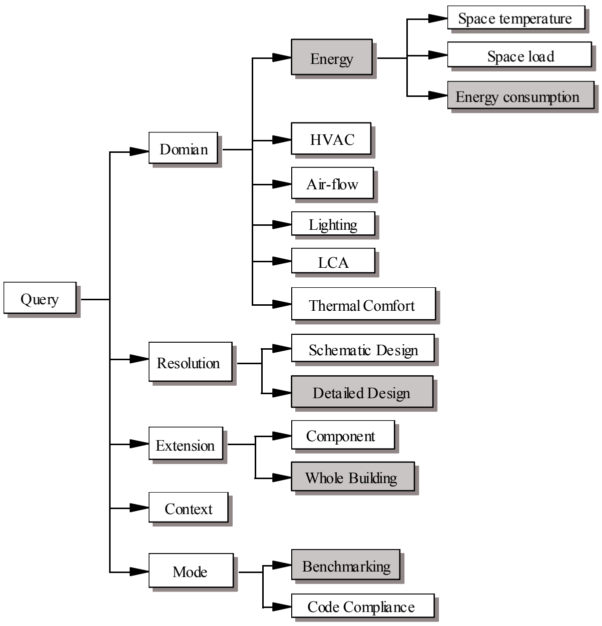

Figure 9 Sample query analysis

CONCLUSION

This paper presented a conceptual scheme to illustrate a query centric approach toward performance assessment in building design. The desired functionalities of the environment are, to interpret user queries and to assemble relevant services and present the answer. The proposed system involves a scalable network of agents, which can respond to performace-related queries posed by architects. The challenges range from formalization of a query language for performance assessment, natural language parsing, query rationalization through mapping to mutli-agent society of services. The prototypes in other fields of study have demonstrated the feasibility of similar approaches. The present paper argued that such approaches may be adapted to the requirments of building performance assessment. Future research will aim to create a working prototype of the proposed system.

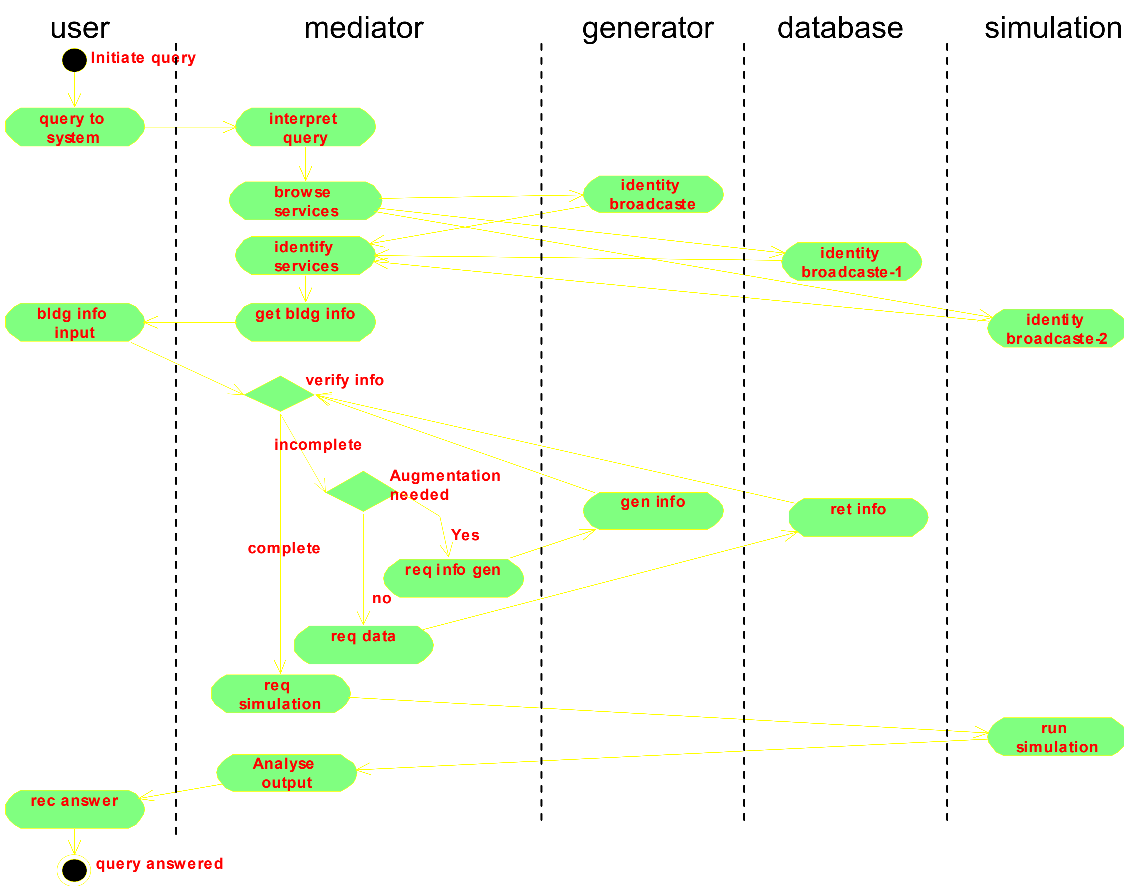

Figure 10 Query flow

REFERENCES

Augenbroe, G., 2003. Design Analysis Initiative. Final Report to DOE. Georgia Institute of Technology, College of Architecture.

Kohout, L.J., Granville, B., Kim, E. 2001. Granular Relational Computing with Semiotic Descriptors using BK-Products of Fuzzy Relations, Computing with Words, Editor Paul P. Wang, Wiley Series on Intelligent Systems.

Liu, J., Autonomous Agents and Multi-Agent Systems, Exploring in Learning, Self-Organization and Adaptive Computation. World Scientific, Hong Kong, 2001.

Mahdavi, A., Lam, K. P., Gupta, S., Wong, N. H., Brahme, R. and Kang, Z. Integrated and distributed computational support for building performance evaluation, Advances in Engineering Software, Volume 33, Issue 4, April 2002: 199-206

Mitamura, Nyberg, Torrejon, Svoboda, Brunner and Baker 2002. Pronominal Anaphora Resolution in the KANTOO Multilingual Machine Translation System, Proceedings of TMI 2002 9th International Conference on Theoretical and Methodological Issues in Machine Translation, Keihanna, Japan

Sycara, K., Paolucci, M., van Velsen, M. and Giampapa, J., 2003. The RETSINA MAS Infrastructure, the special joint issue of Autonomous Agents and MAS, Volume 7, Nos. 1 and 2, July, 2003