

Anas Alfaris

e-mail: anas@mit.edu

Afreen Siddiqi

e-mail: siddiqi@mit.edu

Charbel Rizk

e-mail: charbelr@sloan.mit.edu

Olivier de Weck

e-mail: deweck@mit.edu

MIT Engineering Systems Division,
Building E40-261,
77 Massachusetts Avenue,
Cambridge, MA 02139

Davor Svetinovic

MASDAR Institute of Science and Technology,
P.O. Box 54224,
Abu Dhabi, UAE
e-mail: dsvetinovic@masdar.ae

Hierarchical Decomposition and Multidomain Formulation for the Design of Complex Sustainable Systems

Designing a large-scale complex system, such as a city of the future, with a focus on sustainability requires a systematic approach toward integrated design of all subsystems. Domains such as buildings, transportation, energy, and water are all coupled. Designing each one in isolation can lead to suboptimality where sustainability is achieved in one aspect but at the expense of other aspects. Traditional ad hoc allocations of design parameter precedence and dependence cannot be used for cases where new (instead of only mature) architectures are to be explored. A methodology is introduced for addressing design problems of complex sustainable systems that is comprised of, on the one hand, a hierarchical decomposition that includes multilevel abstraction and design parameter identification, and on the other hand, a multidomain formulation, which includes parameter dependency identification, design cycle identification and decision structuring, and scoping. The application of the methodology for the design of a new urban development, Masdar City in Abu Dhabi, with over 220 different form and behavior parameter sets is shown. [DOI: 10.1115/1.4002239]

1 Introduction

For creating sustainable complex systems, it is important to take an integrated, multidomain design approach. In large-scale complex systems, a number of subsystems (often also complex and large in their own right) interact and function together. For instance, in a city, the building, transportation, energy, and water systems are different complex systems (or domains) that work together so that the city can function as a whole. If it is desirable for sustainability to reduce emissions, decrease waste, and resource consumption, then it becomes imperative to undertake an inclusive approach across all the domains of the system. Otherwise, if each domain is designed independently to conform to a set of sustainability standards, it may very well be that sustainability requirements in one domain conflict with another and that the overall system is less sustainable than it could be. For example, a very stringent requirement for water recycling and reuse might entail very high energy and capital equipment requirements that are in conflict with a low environmental footprint. The overall behavior of the large-scale system is then far from ideal or perhaps in some cases could even be worse [1].

This problem of “balancing” the design in a complex system has been the subject of significant attention and research in the past few decades [2]. Much of the work has been focused on systems such as spacecraft, aircraft, and buildings [3,4]. In this work, however, we address systems that can also be referred to as system of systems (SoS) where each subsystem is a different, fully functioning system in its own right [5]. Specifically, we look at the case of designing a future city where its building, transportation, water, and energy systems have to be designed so that the city has minimum emissions and high efficiency in the use and consumption of resources (such as that of energy, water, and other materials). In such problems, it is critical to balance the design of all the systems so that the sustainable design of one does not impose unnecessary requirements on the other.

When all the systems are treated at a similar level of detail and their inter-relationships explicitly understood, it would be possible to identify the key parameters that collectively provide the greatest influence on the design. The need for eliciting the key driving variables across the systems (which will also be called domains in the context of our work) is fundamental if a systematic approach toward a truly sustainable design is to be adopted. This work of parameter identification and modeling of coupling relationships precedes what is typically understood as mathematical modeling and optimization.

In an integrated, multisystem (domain) design approach, the challenges of scale and scope are prominent. If one is to model each system in detail, then the design space quickly becomes prohibitively large when multiple systems/domains are included. Another problem is that of the design parameter hierarchy. It is often the case that an ad hoc approach is used in defining a top level system whose design then dictates the design of the next system and so on. In some instances, tradition and routine also drive the hierarchy of choice in terms of which system is designed (or its variables set) first and so on. This is usually acceptable when the system design and architecture are mature and the new design is not fundamentally different from experience. However, for cases where it is permissible and, in fact, desirable to explore completely new architectures, it can often be unclear as to how the hierarchy of the design parameters should be set. This difficulty is only compounded with the complexity and size of the problem being addressed, and the relationships that exist between parameters across different domains.

A simple example of a hypothetical building system can be used to illustrate this discussion. It is assumed that a new building is to be designed in which its municipality-supplied fresh water consumption is to be significantly reduced. This problem is motivated by a case described in Ref. [6]. The water reduction is achieved through re-use of treated gray-water and high efficiency appliances and fittings. The building can have three types of spaces (zones): offices, retail areas, and restaurants. Each type has its own requirements for water and produces used water (i.e., gray-water) of different quality. For sustainability, it is also desired that all the energy needs for water treatment within the

Contributed by the Mechanisms and Robotics Committee of ASME for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received January 2, 2010; final manuscript received July 6, 2010; published online September 16, 2010. Assoc. Editor: Steven J. Skerlos.

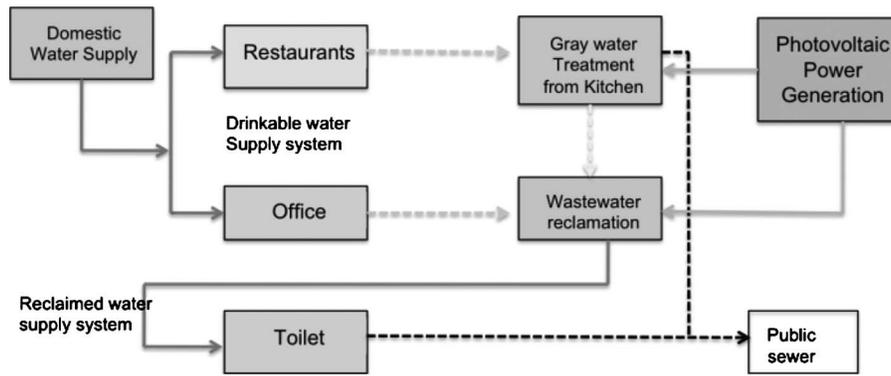


Fig. 1 Sustainable building design with water reuse and treatment with clean energy

building are fulfilled through electric power generated from photovoltaic systems installed on the building's roof. It is also assumed that the size of the building, the floor area mix (in terms of office area, retail area and restaurant area), the water treatment options, and re-use quantities are open for design decisions. Figure 1 shows a simple schematic of the system under consideration.

There are three domains: building, water, and energy in this problem that are closely inter-related. Setting the design parameters for one will impose requirements for the others. For instance, if one is to set the building zone mix (in terms of floor area of each type), then there will be implications for the required water supply and therefore treatment units and electricity generation and so on. Similarly, if one were to set the power generation capacity first, then there will be implications on quantity of treated water which in turn will dictate amount of water that can be re-used. In summary, if the design space of the problem is relatively open (i.e., most of the parameters are not constrained a priori), then it is not immediately obvious which design decisions should be taken first. It is, however, clear that an ad hoc approach toward setting the parameter hierarchy (i.e., deciding the order in which parameters are given design values) can lead to suboptimal solutions. Despite advances in all-in-one approaches to design optimization when the number of design parameters exceeds a few dozen, the problem becomes too large for convergence to be achieved in useful time.

For large-scale systems, where new architectures can be explored, and a number of different domains (each equally important and also complex) are intimately related, it is vital to take a systematic approach. Our work describes a systematic framework applied in a consistent manner across all constituent systems of a larger complex system so that the structure and relationships of the design parameters can be organized. This organization then helps toward structuring the flow of design decisions.

Our methodology consists of first decomposing the system by creating an abstraction of the design problem levels through the use of object-process methodology (OPM) [7]. The object-process diagrams (OPDs) of the overall problem and then of specific systems to be designed are created in a consistent manner at the same level of decomposition. This results in the identification of the most relevant objects (form) and process (behavior) parameters at the same level of detail across all systems and allows for meaningful construction of dependency relationships. The form and behavior parameters of each system (or domain) are then formulated and assembled in a metamatrix, the multidomain matrix (MDM), that consists of the intradomain design structure matrices (DSMs) [8] and interdomain domain mapping matrices (DMMs) [9]. The completed MDM is then partitioned and clustered using reachability matrices. The result is a sequencing and grouping of the parameters so that an ordering of design cycles (smaller sub-problems within a larger design problem) becomes apparent. Each design cycle is then further analyzed to allow for validating and

structuring design decision hierarchy. A traceable and justified order of design decisions and cycles for the multidomain, complex system design problem is thus created.

The methods, of OPM, DSMs, and reachability matrices, discussed in this paper are known and separately developed in literature. This work, however, brings them together to show how a systematic and traceable methodology can be constructed for structuring large-scale design problems. While the proposed method is applicable for any relevant design problem, it can be especially useful in the context of sustainable design. As discussed earlier, an integrated cross-domain approach is needed to create overall sustainability in engineered systems. Cross-domain integration, however, creates a larger design space that needs to be systematically handled. The methodology proposed offers a structured way for systematically approaching such large-scale design problems.

In Sec. 2, we describe the details of our method in a stepwise fashion, and then in Sec. 3, we discuss its application to an urban development project currently ongoing in the UAE. In Sec. 4, we discuss the strengths and limitations of our approach and provide a roadmap for future work.

2 Methodology

Our proposed methodology for systematically addressing a complex, multidomain design problem essentially consists of the following steps (depicted in Fig. 2):

1. Hierarchical decomposition of each system (domain), which includes the following
 - a. multilevel abstraction
 - b. identification of form parameters (FPs) and behavior parameters (BPs)
2. Multi Domain Formulation which includes the following:
 - a. identification of information flow (dependency) between parameters
 - b. design cycle identification through sequencing and clustering of the parameters
 - c. decision structuring and scoping of design cycles through higher order parameter dependency analysis

2.1 Hierarchical Decomposition. We use the term "decomposition" to refer to the act of breaking a large problem into a set of smaller problems or elements. Considerable literature exists on the subject of design decomposition. The principal research in this literature was initiated with the work of Alexander [10] on using networks for representing and decomposing design problems according to customer needs. In system theory a well-established decomposition approach is the *what-how* method. The "what" is a problem-oriented question that conveys a goal that in turn refers

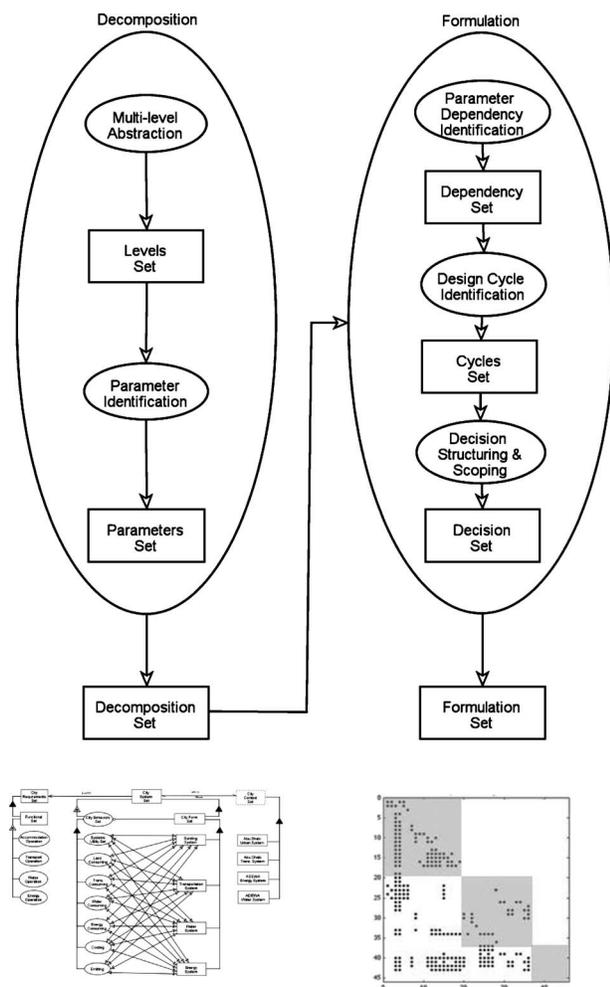


Fig. 2 Methodology overview: hierarchical decomposition and multilevel formulation

to the system's function. On the other hand, the "how" is a solution-oriented question that conveys how the objective is met and in turn refers to a system's architecture. Axiomatic design [11] is an example of a system decomposition method that implements the what-how paradigm where the *what* is represented by functional requirements and the *how* is represented by design parameters.

In our proposed methodology, we will use a systems modeling language to represent our design decomposition levels and parameters. Several system modeling languages have been developed for general-purpose systems, each with its own semantics and notation, these include unified modeling language (UML), systems modeling language (SysML), and the object-process methodology (OPM).

In our methodology, we will focus on OPM because it provides us with a framework for formally defining *whats* and *hows* and features a concise set of symbols that form a language for expressing the system's building blocks and how they relate to each other both formally and behaviorally.

Building blocks in OPM include objects and processes. Objects are what a system or product is (physical or informational), while processes are things that transform objects. Processes can transform an object in three different ways: by creating it, by destroying it, or by affecting it in some way. In OPM objects and processes together describe a system's function, form, and behavior in a single, consistent, domain-neutral model. In OPM graphical nomenclature rectangles refer to objects, while ellipses refer to processes. OPM uses two types of links to connect entities with

each other: structural links and procedural links. Structural links relate objects to other objects and processes to other processes and can express static relations between pairs of entities. Procedural links, on the other hand, are links used to connect entities to describe the behavior of a system. They link an object to a process and can indicate a change in the state of the object [7].

2.1.1 Multilevel Abstraction. According to Eggert [12], the function of a system is what it is expected to perform. Function describes the rationale behind a system existence, the intent for which it was built, the purpose for which it exists, and the goal it serves [7]. When dealing with functions, we are concerned with what the system does and why it does it and not with how it does it.

The system's architecture is the general system's form-behavior combination that helps it achieve its function. It represents the thing that is eventually implemented and operated in a solution specific domain. Form, according to Crawley [13], refers to the physical or informational embodiment that exists or has the potential to exist. On the other hand, the behavior of a system describes how an object operates to achieve its function. Function and behavior are not synonymous. While function is derived from the system's goal; behavior is how the system behaves to achieve the system's function. Performance of a system therefore becomes an attribute of a system's actual or expected behavior that measures its effectiveness.

In the general context of designing, functions, represented through requirements R (where R is a set), are to be transformed into a system form F that can produce the required functions. A set of design requirements R might not have a unique form F solution but a multiplicity of solutions. Two different forms can achieve the same function through different behaviors. The behavior of the system B can then be compared with and evaluated against the functional requirements R to determine if the synthesized form achieved the required functions.

Form-behavior decomposition usually refers to hierarchically related subsystems presented schematically as a pyramid whose top is the higher-level system and whose base is the lower level subsystem. Subsystems may correspond to form components and can also correspond to system behaviors. System decomposition relies highly on system designer's experience. It is not typical that designers persistently follow a top-down decomposition process to the level of single parameters. They usually iterate between the upper and lower system decomposition levels according to what they can potentially learn within the process about the implications of some of their architectural decisions and any legacy elements to be incorporated.

In our methodology we propose an OPM template that provides a framework for decomposing a systems *whats* and *hows* with minimal ambiguity and at different levels of granularity. The level of abstraction will be determined based on (a) the detail needed to assist decisions made by engineers and urban planners and (b) the manageability and number of elements. The template describes a design state D_s at a particular point in time and maps the system architecture (form-behavior combination) to the system's requirements and identifies its relation to the context in which the system will operate. Using the proposed OPM template, new form or behavior parameters can be added, modified, or even omitted as the design evolves. There are three main sets that capture a design state, these include the requirements set R , the context set C , and the designed system set S (Fig. 3). Both R and C provide input to the system set S parameters. The nature of these sets and the number of entries in each are closely coupled with the purpose being modeled and the complexity of the system being formulated. Using the three design sets, a descriptive measure of the design state D_s can be provided at any given instant in time $D_s = \{R, C, S\}$.

2.1.2 Parameter Identification. Within the proposed template, the requirements set R is the cornerstone of the systems design

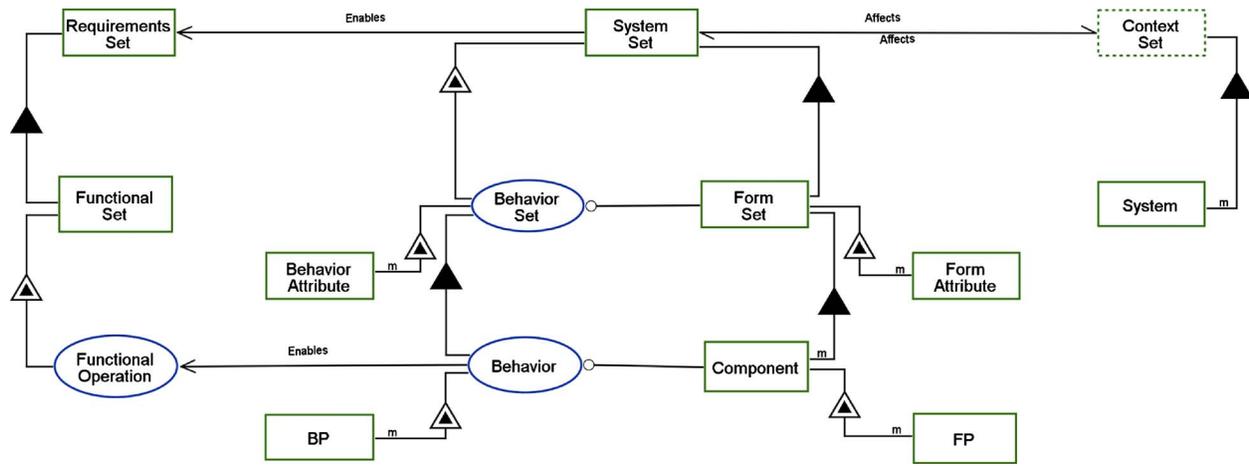


Fig. 3 OPM decomposition template

and engineering process. Buede [14] assumed four categories of requirements: input/output, technology and systemwide, trade-off, and test requirements. The context set C is the set of entities that interact with the systems through its external interfaces. The context entities can impact the system as a form of input and be impacted by its outputs. The context set could include operands in the context environment, external systems, or the system users. The system design set S includes the system's form set S_f as well as the system's behaviors set S_b .

Identifying design parameters is the main objective of our decomposition. We consider two main classes of design parameters: FPs and BPs. FPs describe the systems attributes, components, and relationships, while BPs describe behavior characteristics that are derived or expected from the system's form. FPs are essential to the synthesis activities later in the modeling phase, while BPs are important for the analysis activities. Each component can have one or more form parameters FP that describe the components attributes. The system form executes certain system behaviors. The system's behavior set includes the anticipated system behaviors. Each anticipated behavior can have one or more behavior parameters BP that measure the behavior characteristics. Each of these sets should include elements at the level of granularity available at that level of the design.

To illustrate the use of this template, consider the design problem introduced in Sec.1. The problem can be abstracted into two levels. For brevity we will focus here on level 1. The context set would include the city urban system, the city water system, and the energy system. The requirements set would include functional requirements such as accommodation, water and energy operations, and requirements. The system set would include high level subsystems, as well as high level system behaviors. The form set would include the building spatial system, water system, and energy system. The behavior set would include the high level utility, area consuming, water consuming, energy consuming, and costing and emitting behaviors (Fig. 4). In the second level, we have more detailed parameters, for example, the spatial system, for example, includes several FPs, such as the zone mix, that will affect several BPs, such as water demand and building cost.

2.2 Multi Domain Formulation. Formulation can be defined as the act of systematized expression. It can imply an arrangement of a group of elements in a prescribed manner or for a specific purpose [15]. In the process of designing a complex system, it is necessary to structure the information and different parameters extracted at the decomposition phase, as well as describe and plan the sequence of applications and interactions.

Our proposed method views the design of a large-scale complex system as an interaction between parameters within a set of design cycles. Design cycles and iterations are part of any design process.

They can be defined as the repetition of activities to improve an evolving design [16]. A design cycle can be modeled by tracing design information flow between different design activities and parameters.

2.2.1 Parameter Dependency Identification. Eppinger [17] identified three types of flow dependencies: serial (dependent), parallel (independent), and coupled (interdependent). Three types of parameter dependencies can also be identified: These are form to form, behavior to behavior and form to behavior. Form to form dependencies identify which FPs provide information to a particular FP. Behavior to behavior dependencies identify which BPs provide information to a particular BP. Form to behavior dependencies identify which FPs provide information to a certain BP.

To understand the design process and the structure of information flow, different representations can be used. One of such representations is the DSM. The DSM was first introduced by Steward [8], and since then has gained widespread use in design and organization of complex systems. The DSM records relationships between components, tasks, parameters, or other entities that can be used to describe a system. In our methodology, DSMs are used to capture flow of information between design parameters and to then elicit the inherent hierarchy that exists between them.

For a given set of N parameters, the DSM is assembled by listing them along the columns and rows of an $N \times N$ matrix. The matrix is populated by putting a 1 in the i th row and j th column if parameter j provides information to parameter i . If there is no informational dependency between a pair of parameters, their corresponding matrix element (or cell) is set to zero. The DSM thus created shows information flow (i.e., in the example above, information flows from parameter j to i).

In our methodology, both the requirements set R and the context set C identified in the decomposition provide input to the system set S parameters. Therefore, the focus here will be on S and the list of FPs and BPs obtained from the results of decomposition of the systems, as described in Sec. 2.1. In a multidomain problem, the DSM concept is extended to that of a DMM in which the relationships between parameters of different domains are also captured. A metamatrix called the MDM can then be assembled, in which the DSMs are ordered diagonally, while the DMMs are placed in off-diagonal positions.

For instance, consider the simple design problem introduced in Sec. 1. Suppose that after the decomposition process, as described in Sec. 2.1, a set of design parameters are obtained for each domain. Figure 5 shows a sample MDM for this case. The intradomain parameter dependencies are described in the DSMs (colored blocks along the diagonal), while the interdomain relationships have been identified in the off-diagonal blocks in DMMs for each

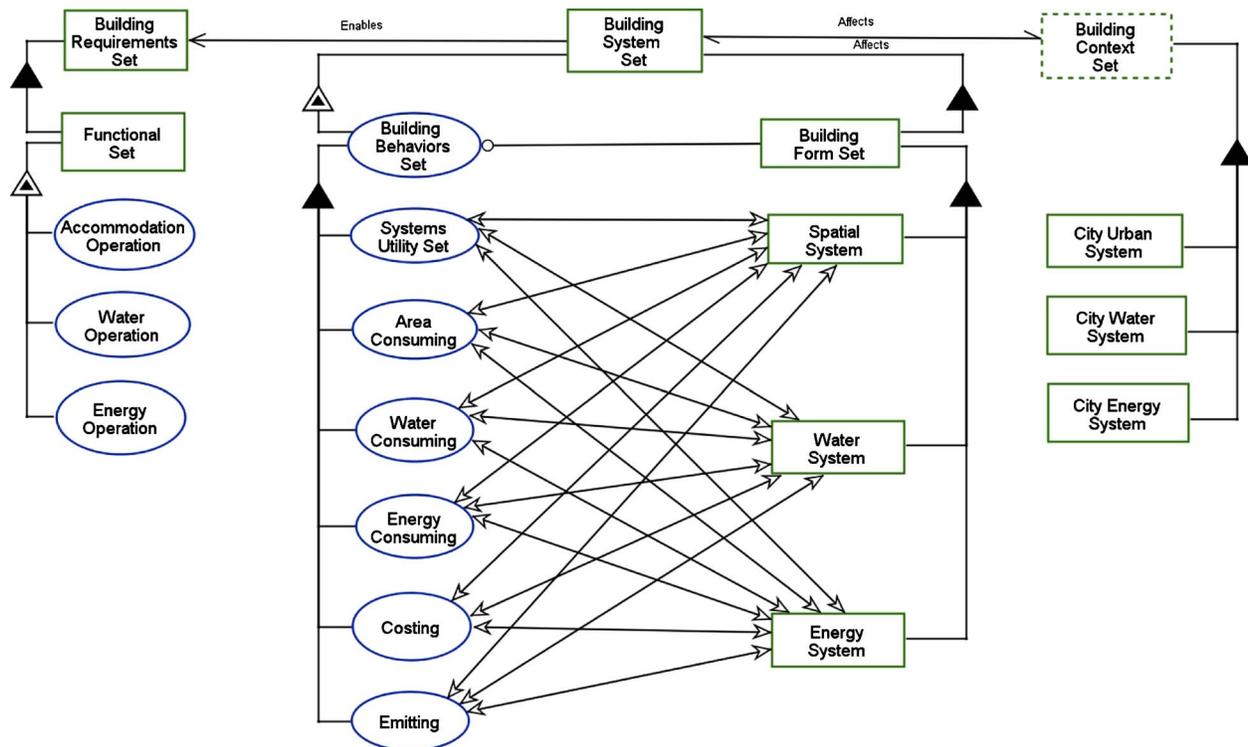


Fig. 4 OPM decomposition for sustainable building design example

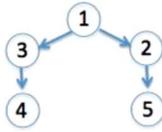
domain pair. The parameter prefixes W, E, and B (as can be seen in the figure) indicate their domains of water, energy, and building as was introduced in Sec. 1. For instance, parameter W1 is the building water supply volume per day, E1 is waste-water treatment energy consumption, and so on.

It is interesting to observe from the MDM that the three domains are not isolated (the 1s in the white DMM blocks indicate presence of interdomain relationships). If each domain had been

independent, its design process can proceed without needing much interaction from other domains. However, in the case of inter-relationships if other systems (domains) are neglected up-front, then there may be a need to perform more iterations (than necessary) to arrive at a feasible solution (which may not be the best), or if one domain is given decision priority over others, then its solution will dictate the design of the others that may lead to overall inefficient and suboptimal design.

	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	E1	E2	E3	E4	E5	E6	B1	B2	B3	B4	B5
W1	1																	1				
W2		1																	1			
W3			1																		1	
W4				1									1									1
W5					1																	1
W6						1							1									
W7							1															
W8								1														
W9									1													
W10										1												
W11											1											
E1												1										
E2													1									
E3														1								
E4															1							
E5																1						
E6																	1					
B1																		1				
B2																			1			
B3																				1		
B4																					1	
B5																						1

Fig. 5 A MDM for the sustainable building design example



$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 6 Dependency relationship between five elements in graphical and DSM format

2.2.2 *Design Cycle Identification.* While a good quality MDM can be very valuable in itself, a logical and strategic resequencing of the parameters in the matrix is an essential next step. This re-ordering of rows and columns (also referred to as *partitioning*) allows for identifying the inherent feedback that may exist in terms of information flow between a set of parameters. A feedback exists when there are 1s in the upper diagonal of the matrix (indicating that an upstream parameter needs information from a downstream parameter). A lower triangular matrix implies no feedback (thus computation of a design can proceed in a completely sequential manner). Ideally, once a MDM has been partitioned, only the inherent feedback in the information exchange structure of the system remains, while any other feedback effects that arise purely due to an arbitrary ordering of the parameters in the matrix are removed. The result is a structured hierarchy of design parameters in which only the inherent coupling and feedbacks remain and the decision flow of the design process starts to become apparent.

Another important step is *clustering* of the MDM. Clustering produces modules, i.e., it produces an ordering such that elements or parameters that are coupled or have higher degree of interaction within them (as compared with the rest) are sorted out in groups. This technique helps in managing complexity and helps in organizing process flow [18].

In our framework, sequencing and clustering identify groups of parameters that constitute a smaller design problem; i.e., they form the design space of a design cycle within the larger complex problem. The concept of reachability matrices [19] is used to identify important parameters belonging to a design cycle. When binary matrices (such as a binary DSMs) are raised to different powers (using Boolean multiplication) the reachability or paths that exist between elements can be found. This idea has been the basis of element visibility in complex systems [20,21]. A simple system, as shown in Fig. 6, can be used to illustrate the concept.

The DSM of the system shown in Fig. 6 is given as matrix A . The square and cube of this matrix is

$$A^2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad A^3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The matrix A^2 essentially shows (with ones in the first column and in the fourth and fifth rows) that there is a path, of length 2, that exists between element 1 and element 4 (along path 1-3-4) and 5 (along path 1-2-5). Matrix A^3 being entirely zero essentially means that there is no pathway of length 3 between any two elements in the system.

In a system with N elements, the maximum path length that can exist is $N-1$. Thus, for a system with N elements, a visibility matrix V maybe defined as [21]

$$V = \sum_{n=1}^{N-1} A^n$$

The visibility matrix shows the direct and indirect dependencies of all possible path lengths between any two elements. For the system shown in Fig. 6, the V matrix is

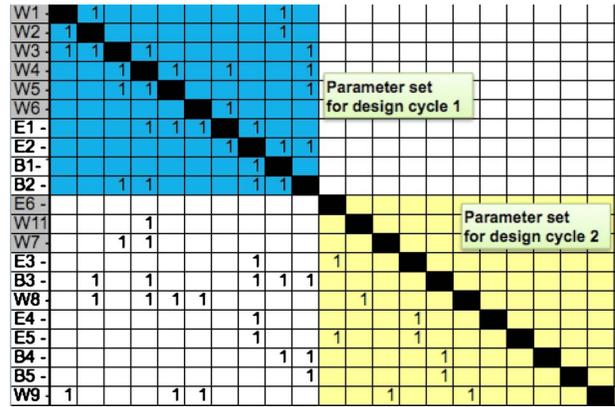


Fig. 7 MDM with design cycle grouping—note mixing of parameters from various domains in cycle 1 and 2

$$V = \sum_{n=1}^4 A^n = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

It can be seen that element 1 has accessibility to all other elements (2–5). In the context of design parameters, the DSM in matrix A shows that parameter 1 directly influences (provides information to) parameters 2 and 3. However, the DSM does not readily show the indirect dependence that parameter 1 has on 4 and 5. The higher successive powers of A and the V matrix collectively provide richer insight into the influence chains and can be useful in determining critical and related parameters (within a design cycle). In this case, for instance, it is clear that parameter 1 should be treated (or decided) carefully since its impact cascades through every other parameter.

Using this reachability matrix concept design cycle blocks can be identified. It is important to observe that loops are found within blocks (design cycles) rather than between blocks. Once the cycles and their parameters are determined, the original Boolean MDM is rearranged to show the separate design cycle blocks. This type of clustering can be used to determine the order in which design cycles can be solved. A cycle is thus identified based on having both FPs and BPs with FPs generally preceding BPs. Each cycle block represents a collection of FPs and BPs that need to be determined through mathematical modeling. The values of the parameters are then fed into the subsequent design cycles. Once all the cycles have been computed, a full solution of the problem is obtained. In essence, this approach produces an ordered series of subproblems in which the scope of each problem (i.e., collection of parameters) gets selected based on the inherent information and dependency structure of the larger design space. It is possible that in a larger multidomain context, the smaller subproblems may consist of parameters from different domains (mixing). It should be noted that in traditional practice, this approach is typically not adopted and each domain (or system) is designed independently and design iterations are carried out to simply produce interdomain functional feasibility.

Figure 7 shows a sample partitioned and clustered MDM obtained from the original matrix given in Fig. 5. The set of 22 parameters have been grouped into two design cycles. The first cycle has ten parameters from all the three domains (note the parameter prefixes). Also there is a group of form parameters (colored in gray) and a group of behavior parameters (in white). The second cycle has a similar grouping and can be executed after the first cycle has been completed.

2.2.3 *Decision Structuring and Scoping.* The partitioned and

sequenced MDM provides a first estimate of the design cycles in terms of parameters. In a complex system, however, it is important to take a deeper look at the parameter relationships so that each design cycle can be better structured and executed within.

Further insight regarding parameter influence and dependency can be obtained by computing for each parameter, the number of parameters it influences, and the number of parameters it depends on. This is similar to the notions of in-degree and out-degree in graph theory. More specifically, one can define

$$\pi_j = \frac{1}{N} \sum_{i=1}^N a_{ij}$$

$$\delta_i = \frac{1}{N} \sum_{j=1}^N a_{ij}$$

where π_j is an indicator of the fraction of total elements to which element j provides input (normalized column sum), δ_i is the fraction of total elements on which element i depends (normalized row sum), and a_{ij} is an element of a matrix that can be the DSM, a power of the DSM, or the V matrix. The term “fan-out visibility” and “fan-in visibility” for π and δ has also been used in literature [21].

The DSM, its powers, the V matrix, and values of π and δ collectively serve to provide an information basis for guiding the structure and scope of decisions in a design cycle. For a simple illustration consider again the DSM of the design problem shown in Fig. 5. Using the water domain DSM (upper left 11×11 portion of Fig. 5), the parameter with the highest π was found to be W4 (which was “volume of treated water per day”). Next, using the V matrix, the highest value of π turned out to be for W1 (volume of city water supplied per day) instead. So if only first order interactions are accounted for, W4 is the most influential parameter. However, when higher order interactions are also factored in, the influence ranking changes and W1 is found to play an even more important role overall. Both of these parameters belong to the first design cycle, as shown in Fig. 7, and should be treated with due deliberation keeping in mind the impact they will have on the rest of the parameters.

3 Application: Masdar City

We have applied our methodology to the ongoing urban development project of Masdar City in the outskirts of Abu Dhabi in the United Arab Emirates (UAE). The city master plan has been designed by the British architectural firm Foster+Partners and is targeted to be a sustainable, zero-carbon, zero waste community that will depend on renewable energy sources [22]. The city of Masdar, with a planned area of 6 km^2 , will be located 17 km south-east of Abu Dhabi. It will house 50,000 people, 1500 businesses, and a technical university (Masdar Institute of Science and Technology). It is also expected that more than 60,000 workers will commute to the city daily [23]. The project, initiated in 2006, is estimated to cost US \$22 billion and is targeted for completion in 8 years [23]. An aerial view of the envisioned Masdar City at completion is shown in Fig. 8.

3.1 Masdar Hierarchical Decomposition. For decomposing Masdar City, the template introduced in Sec. 2.2 was used.

3.1.1 Masdar Multilevel Abstraction. Briefly, the decomposition is composed of three sets; a requirements set R , a context set C , and a city system set S , which affects and is affected by the city context, in order to meet the city functional requirements. The same template will be used for all lower level decompositions. In the Masdar application, three decomposition levels were required to reach the described design state. At each level, we will identify form components and behavior processes that are relevant for the design of Masdar City. The OPM approach discussed earlier will be used to represent the decomposition.



Fig. 8 Grid-based layout of Masdar City at completion, approximately 2016. The enclosed area is approximately 6 km^2 and houses 50,000 people. Design of the city to meet high level sustainability targets is an ongoing design challenge.

3.1.2 Masdar Parameter Identification. At the highest level (Fig. 9, L1), two types of behavior were identified: common behaviors, seen in all subsystems/domains (with different impacts), and specific behaviors, which are particular to a certain system. It is also important to note that behaviors of subsystems will aggregate to feed behavior of higher-level systems. In our level 1 decomposition of Masdar City, we focused on the system design set of the supersystem level at the city level. Within the context set, the city will interface with existing systems including the Abu Dhabi Energy and Water Authority (ADEWA) power grid, ADEWA water distribution system, ADEWA sewage system, the Abu Dhabi transportation network, which includes the road network, and the Light Rail Train (LRT). The requirements set mainly includes the provision of sheltering/accommodating, transporting, providing energy/water, for city residents [22]. Within the system’s design set, the form set includes four subsystems; these are the building system, transportation system, energy system, and water system. These subsystems have several common behaviors that are important to evaluate when designing the city sustainably, which include land use, transportation use, energy use, water use, cost, and emissions, as well as the system specific behaviors. Several dependencies exist between the different subsystems and sub-behaviors which have to be considered at lower level decompositions.

In the level 2 decomposition (Fig. 9, L2), we provided more detailed views of each of the four subsystems discussed in level 1. A separate OPD was designated for each of them. For brevity, the level 2 decomposition discussion will be restricted to the building subsystems. Similar to level 1, level 2 contains three sets. Considering the building system first, the context set will include aspects such as natural conditions, regulations and codes, and geographic location of the city. All these external factors will influence FPs and expected BPs.

The requirements set will consist of aspects such as zone dependencies and building area requirements. The city will include different zone types such as the university located within a special economic zone, an associated commercial zone, light industry zones, and residential accommodations. The requirements also necessitate a design with a high quality of urban spaces and buildings within which a sense of place is created and through which a community will prosper and a pleasant environment will be formed making the city a place where people want to live in [22]. The building system OPD consists of two main components and 14 behaviors. The two components we identified are type and network. As for behaviors, they are similar to the ones included in level 1 but are split into two behaviors that capture the impact of

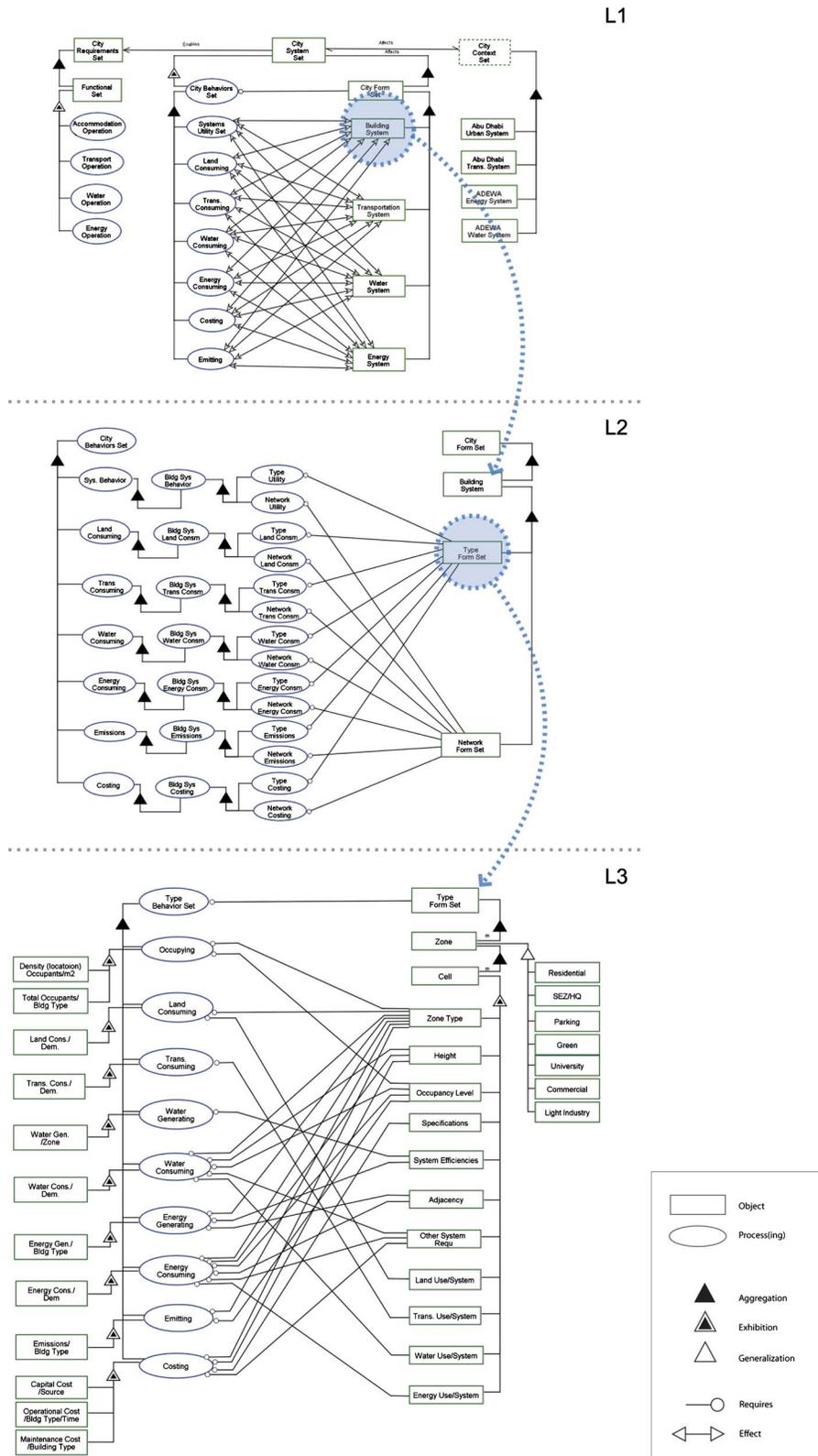


Fig. 9 Hierarchical decomposition of the Masdar Building System: L1: city level, L2: domain level, and L3: component level

type and network components.

At level 3, we decomposed the type and network components identified at level 2 (Fig. 9, L3). The context and requirements set at this level align with ones at higher levels but are more specific and relate to the subsystems themselves. Level 3 decomposition of

the type form set within the building system consists of several zones, such as residential, special economic zones/headquarters SEZ/HQ, parking, and green areas [22]. Each zone consists of many cells that require several specifications for land, water, transportation, and energy use as well as other FPs specific to

Table 1 FPs and BPs for Masdar subsystems (total: 228)

OPD Level 3			
System	Type	No. of FPs	No. of BPs
Building	Mode	11	14
	Network	12	8
Water	Mode	19	10
	Network	17	13
Energy	Mode	21	10
	Network	14	13
Transportation	Mode	18	16
	Network	21	11

subsystems. The type behavior set includes several behaviors such as costing, energy consuming, energy generating, emitting, occupying, land consuming, transportation capacity consuming, and constructing. Each of these behaviors exhibits one or more BPs. For example “costing” exhibits maintenance cost per building type, operational cost per building type, and capital cost per building type. In turn, costing requires information from FPs such as specifications, height, and zone type.

The generated OPDs summarize the information identifying which FPs are required, as well as which BPs are exhibited. A similar level of detailed decomposition was performed for all other three systems. Table 1 represents the number of FPs and BPs for each subsystem.

3.2 Masdar Multidomain Formulation. Here we are concerned with the design processes formulation and design cycle identification for the design of Masdar City. To achieve this, we will use the DSM methodology discussed in Sec. 2.2.

3.2.1 Masdar Parameter Dependency Identification. After identifying the FPs and BPs in the decomposition stage, we will then identify the dependencies between the parameters. Unlike the top-down approach taken in decomposition, we will use a bottom up approach in our formulation. We will start first by populating the individual system DSMs after which we will populate the interdomain DMMs to construct the full MDM. The MDM is

populated with the information identified in the decomposition, which is based on modeling needs and expert judgment of engineers and architects who have particular domain knowledge and experience.

Given the Masdar decomposition discussed in Sec. 3.1, the MDM will be populated with the FPs and BPs identified within the OPDs. For brevity, we will focus our discussion on the formulation of the building and transportation systems (Fig. 10). These two domains were selected because they are highly dependent on each other based on the correlation between city layout and transportation needs. In the MDM of Fig. 10, the building domain FPs (colored in dark gray) are followed by the building domain BPs (colored in white) inside the DSM(B). Similarly the transportation FPs (colored in light gray) are followed by the transportation BPs (colored in white) in the DSM(T). Figure 11 summarizes the 16 different DSMs and DMMs within the MDM prepared for both systems. We can observe from the MDM that the two domains are not isolated. The intersystem DMMs (gray blocks) have several dependencies although lower in magnitude than the intrasystem DSM dependencies of each system (white blocks). Typical intrasystem densities are on the order 0.1–0.3 while intersystem dependency ratios are on the order 0–0.1.

3.2.2 Masdar Design Cycle Identification. After formulating the dependencies within the MDM, we can partition and cluster the completed MDM to identify inherent feedbacks. For our partitioning and clustering, we used the algorithm developed by Cho [24]. The algorithm is based on the partitioning method developed by Warfield [19]. The method uses Boolean matrix manipulation and structures the parameters into hierarchical levels. Similar to Warfield, the algorithm uses a reachability matrix to identify parameters belonging to a certain hierarchical level.

Figure 12 shows the unsequenced DSM(B) of the building system (which will also be referred to as matrix B) on the left. In terms of parameter influence or importance, from simple inspection of the matrix, it can be seen that there are two parameters whose columns are mostly full. These are parameter BS-1 and BS-12 (which are “cell type assignment” and “zone type assignment”). The almost full columns indicate that these parameters directly serve to provide inputs to most other parameters. The associated V matrix (also denoted as V_B and shown on the right in Fig. 12) reveals some additional information. The key information is that there are indeed four parameters that play an influential role

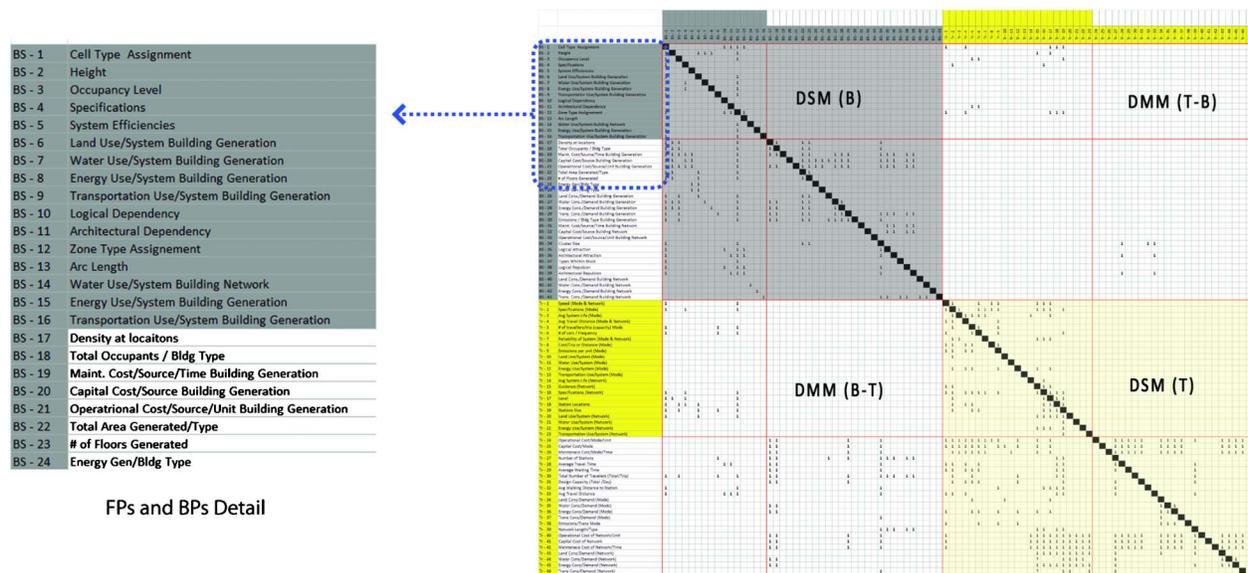


Fig. 10 Building and transportation MDM for Masdar City

	FP1	BP1	FP2	BP2
FP1	# Dep.=32 Size=16x16 Ratio=0.125	# Dep.= 0 Size=16x27 Ratio=0.000	# Dep.=19 Size=16x23 Ratio=0.052	# Dep.=0 Size=16x24 Ratio=0.000
BP1	# Dep.=80 Size=27x16 Ratio=0.185	# Dep.=101 Size=27x27 Ratio=0.139	# Dep.=0 Size=27x23 Ratio=0.000	# Dep.=7 Size=27x24 Ratio=0.011
FP2	# Dep.=31 Size=23x16 Ratio=0.084	# Dep.=0 Size=23x27 Ratio=0.000	# Dep.=98 Size=23x23 Ratio=0.185	# Dep.=0 Size=23x24 Ratio=0.000
BP2	# Dep.=13 Size=24x16 Ratio=0.034	# Dep.=67 Size=24x27 Ratio=0.103	# Dep.=153 Size=24x23 Ratio=0.277	# Dep.=112 Size=24x24 Ratio=0.194

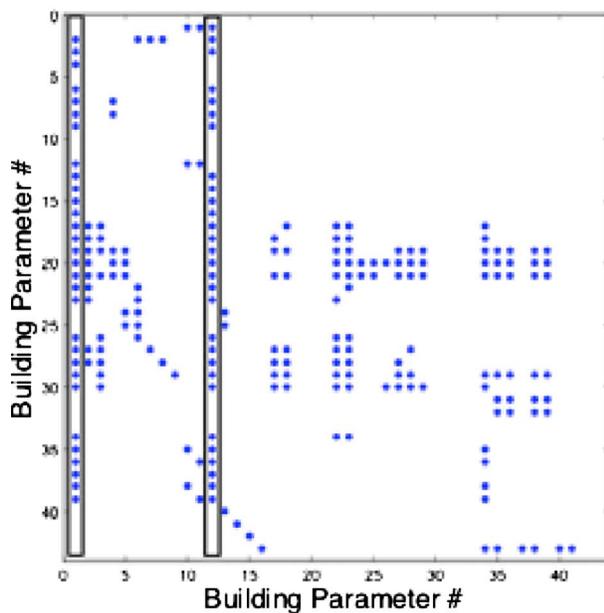
DSM
 DMM

1: Building System
 2: Transp. System

Dep. : Num. of marks in DSM/DMM
 Size : Num. of rows x Num. of columns in DSM/DMM
 Ratio : # Dep. / Size

Fig. 11 DSMs and DMMs within the building and transportation MDM. The Ratio indicates the density of dependencies within each submatrix.

rather than just the two that could be identified from the DSM(B) alone. The additional influential parameters that come to light as a result of computing V are parameters BS-10 and BS-11 (which are “logical dependency” and “architectural dependency”). This means that in terms of direct impact, the most important variables in the building system domain are the cell and zone definitions (for a grid or cell based description of the city’s area). The choices made for these two parameters alone will trickle down to affect almost every other parameter of the building domain directly. Additionally, the decisions for logical and architectural dependency



will also play a key role on impacting other variables.

The result of the partitioning and clustering is a structured hierarchy of design parameters in which only the inherent coupling and feedbacks remain. The decision flow of the design process starts to become apparent. Each generated cluster identifies a possible design cycle with a manageable number of parameters. Figure 13 illustrates the output of our MDM clustering in which three design cycles (shaded blocks) were identified. The partitioned and clustered MDM shows significant mixing of parameters from the two domains, confirming the need for concurrent design. Each design cycle includes a mix of FPs and BPs from both the building and transportation domains. Interestingly, neither the building or transportation domains were dominant in any of the cycles. However, the percentage of FPs in the first cluster was higher than the last cluster and the opposite was true for the BPs. This is because the analysis of behaviors requires the synthesis of form components to be completed first.

As stated earlier, the values of parameters defined within a specific design cycle will feed into following cycles. Therefore, care has to be taken when identifying design cycles due to the effect it may have on following design cycles.

3.2.3 Masdar Decision Structuring and Scoping. In order to get further insights about the parameter dependency relationships identified and the design cycles obtained, additional metrics are computed. A few illustrative results are discussed here.

Figure 14 shows plots of the ratio π/δ (for each of the 43 design parameters of the building system). The ratio of π and δ serves to highlight the parameters in terms of their influence versus dependence. Elements with high values of the ratio are thus those that provide influence to more elements that they depend on and should be given careful attention. The ratios were computed from three different matrices. The first line, with black star markers, shows π/δ computed from the (DSM) matrix B. In the second dot marker line, π and δ are computed from B^2 . This is done so that the relationships that exist at two path lengths between elements get identified. Path lengths greater than 2 are not considered here for simplicity. Furthermore, it can be argued that the effect of the “influence” or visibility will be diluted with increasing path lengths. The third line with triangle markers in Fig. 14 is for π/δ computed using the full visibility matrix V_B for the building system.

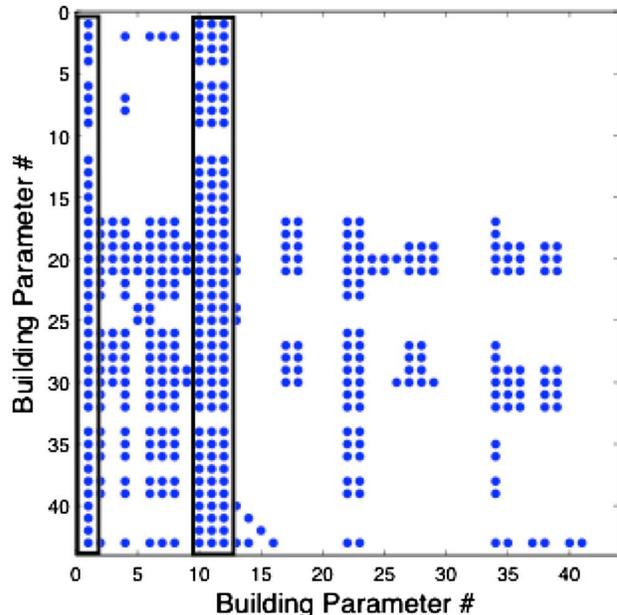


Fig. 12 DSM of building system and V matrix of building system

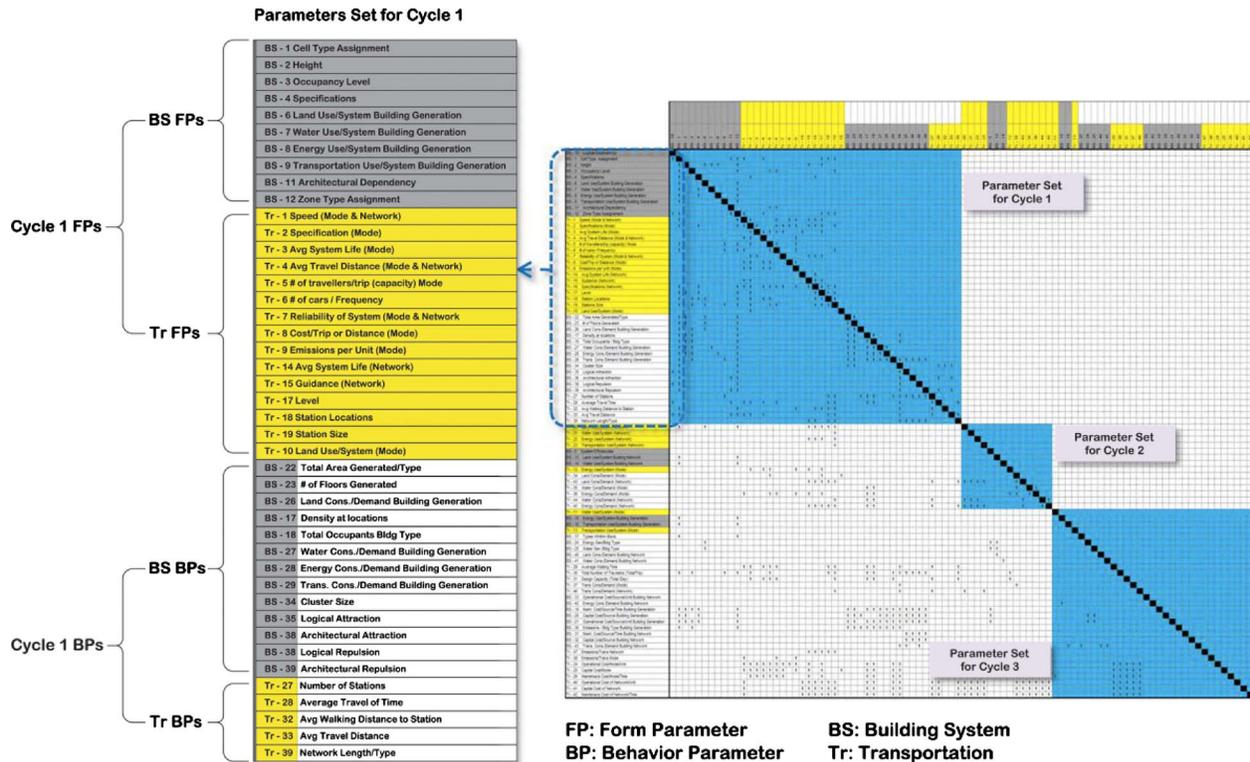


Fig. 13 Masdar City—building and transportation MDM with design cycle grouping

It is interesting to note that there are some parameters in which the ratio across B , B^2 , and V_B simply decreases (e.g., parameters BS-36 to BS-45). But there are some, e.g., parameters BS-1, BS-8 (which is “energy use by buildings”), and BS-12 in which the second-order relationship (B^2) has a π/δ ratio much higher as compared with that obtained from B . This means that there are many more parameters that depend on BS-8 and BS-12 at path length 2 than there are parameters that have direct (unit path length) dependency. We argue that these higher order dependencies are particularly important in complex system design for sustainability.

In order to get a different sense of the impact of each parameter, the π values for each parameter were computed using the B ma-

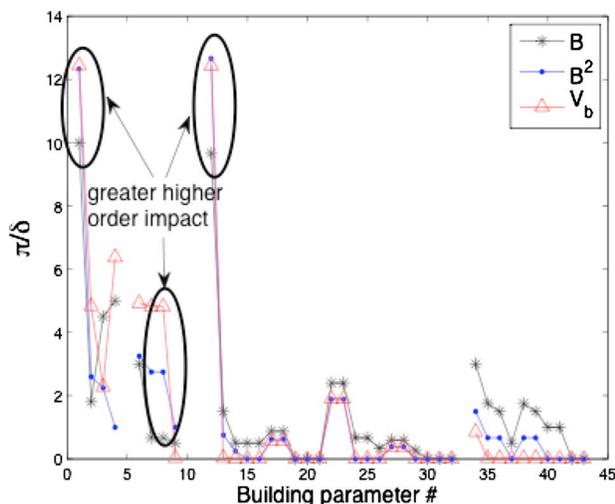


Fig. 14 π/δ ratio for building system from matrix B , B^2 , and V

trix and the V_B matrix and were then sorted by rank. Table 2 shows the top five parameters with highest values of π and also π/δ ratios obtained from B and V_B .

It is interesting to note that all of the parameters that have highest $\pi_{BS-1}(V_B)$ ranking are in the first design cycle (as shown in Fig. 13). Furthermore, it can be seen that BS-1 and BS-12 are far more important than perhaps some others since they have both high direct impact (as can be seen in $\pi_{BS-1}(B)$) and high indirect impact (as can be seen in $\pi_{BS-1}(V_B)$). Furthermore, for BS-1 the π/δ ratio is also among the highest (ranked 5) within the 43 parameters. It has a value of 10 when computed for B and a value of 12.44 when computed for V_B (as can be seen in Fig. 14). This collective information of π and π/δ for both B and V_B means that BS-1 has high first order and higher order impact and that it has low dependency on others. This parameter should thus have a high precedence in the design decisions. An examination of π/δ values for the top ranked BS-33, BS-5, BS-10, and BS 11 shows that these parameters have infinite values since these do not have any dependencies (and thus have zero values for δ). Of these parameters, BS-33 is a behavior parameter, while BS-5, BS-10, and BS-11 are the form parameters. These parameters are thus essentially independent control knobs, and in terms of their impact it can be seen that BS-10 and BS-11 are important (as shown in Fig. 12 and also in second column of Table 2). This information pro-

Table 2 Parameter ranking for highest π and π/δ values

No.	$\pi_{BS-1}(B)$	$\pi_{BS-1}(V_B)$	$\pi/\delta_{BS-1}(B)$	$\pi/\delta_{BS-1}(V_B)$
1	BS-1	BS-10	BS-33	BS-33
2	BS-12	BS-11	BS-5	BS-5
3	BS-22	BS-1	BS-10	BS-10
4	BS-23	BS-12	BS-11	BS-11
5	BS-34	BS-2	BS-1	BS-1

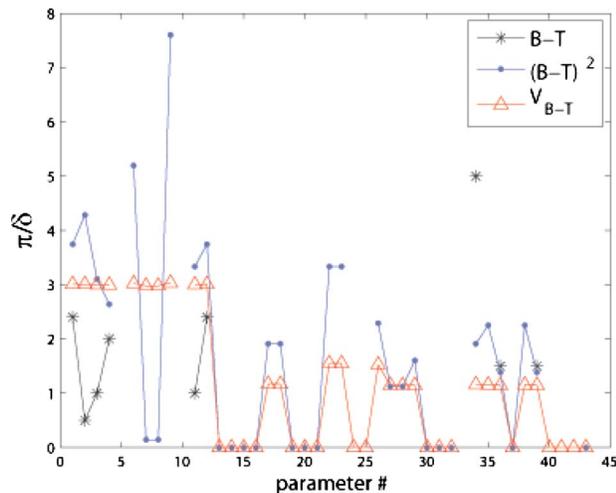


Fig. 15 π/δ ratios for building-transportation DMM

vides guidance in setting parameter precedence rules within design cycle 1 and can also be a basis for making decisions about reducing the feedback (through a process called “tearing” [25]).

Figure 15 shows the ratio of π and δ for each parameter for the building and transportation DMMs. The ratios are computed using the DMMs, their squares, and their associated visibility matrices. The interesting thing to note is that π/δ for almost every parameter is higher for the squared and V_B matrix case (the second and third lines are higher on the y-axis than the black lines with asterisk). This means that the second and higher order effects appear to dominate the influence or impact of the parameters across the domains. In particular, parameters BS-2, BS-6, and BS-9 (all included in design cycle 1) show significant higher order effects. These parameters are “building height,” “land used by building system,” and “transportation use by building system.” This means that when considering the values (through either decisions or computation) of these parameters, their cross-domain impact on the transportation system should be factored in.

4 Conclusions

The methodology presented in this paper is directed toward creating sustainable design solutions through an integrated multi-domain approach. Typically, in many integrated design efforts where a large number of design parameters are to be determined, many key parameters are “set” in an ad hoc manner, thus locking in implications for the subsequent design possibilities. For cases where entirely new solutions are to be conceived, this kind of approach can lead to suboptimal, biased results.

In our methodology, the problem of scale and scope, inherent in large-scale complex system modeling and design, is managed through systemic hierarchical decomposition and a multidomain formulation. The result provides a parameter hierarchy that is reflective of the influence and dependency of the parameters in the problem, and the design cycle structuring allows for organizing the computation and analysis. Using visibility matrices and related metrics, the decisions regarding individual design parameters within the cycles can be better informed. Furthermore, the higher order interactions as they propagate in the DMMs can be very valuable in understanding the design impacts across domains. It should be noted that our proposed methodology is applicable for early stage assessment when detailed models are not available and only a framework in terms of directed relationships between elements/parameters can be constructed (as is the case for MDMs). The method is particularly helpful when design decisions between different domains such as building structures, water, energy, and transportation have to be coordinated within and across.

The current limitations of our approach include a few aspects. First, performing the decomposition process well is not trivial for complex systems, especially when domain neutrality and good quality abstractions are also desired. Second, creating a high quality MDM is an involved process that requires expert knowledge and judgment. Third, the design cycle ordering is based on information exchange alone. It is important to keep in view that a binary DSM based structuring does not differentiate between the intensity and degree of dependence or sensitivity between parameters. Lastly, in using the V matrix for determining higher order element interaction, the magnitude of influence does not get factored in. This type of understanding and insight can only be gained through detailed modeling and simulation in which sensitivity analysis and other systematic means can be employed for determining the impact of change of one parameter on another. The method proposed here is a precursor before a formal design optimization formulation can be set up.

In our future work, we will improve the design cycle determination process through the use of more sophisticated DSMs (that will not be binary). We will also develop more formal algorithms based on visibility concepts but tailored to meet the needs of creating better grouping of parameters for design cycles. In this work, we have focused on discussing how the interactions alone between parameters can be the basis for structuring precedence of design decisions. In future work, the rules for setting parameter precedence and inclusion in design cycles can also be based on how a parameter maybe tied to sustainability metrics or goals. Furthermore, this framework as it is applied across all four domains (of building, transportation, water, and energy) will be used to produce a detailed modeling and design tool for Masdar City. This will help in rationalizing and refining current designs and plans for this large-scale undertaking that can serve as a template for future sustainable developments.

Acknowledgment

This work was supported by funding from a MIT-Masdar Institute of Science and Technology Collaborative Research Grant.

References

- [1] Taber, J., Balling, R., Brown, M., Day, K., and Meyer, G., 1999, “Optimizing Transportation Infrastructure Planning With a Multiobjective Genetic Algorithm Model,” *Transportation Research Record No. 1685*.
- [2] Agte, J., de Weck, O., Sobieszcanski-Sobieski, J., Arendsen, P., Morris, A., and Spieck, M., 2010, “MDO: Assessment and Direction for Advancement—An Opinion of One International Group,” *Struct. Multidiscip. Optim.*, **40**(1–6), pp. 17–33.
- [3] Braun, R. D., Moore, A. A., and Kroo, I. M., 1997, “Collaborative Approach to Launch Vehicle Design,” *J. Spacecr. Rockets*, **34**(4), pp. 478–486.
- [4] Sobieszcanski-Sobieski, J., and Venter, G., 2005, “Imparting Desired Attributes in Structural Design by Means of Multi-Objective Optimization,” *Struct. Multidiscip. Optim.*, **29**(6), pp. 432–444.
- [5] Maier, M., 1998, “Architecting Principles for Systems-of-Systems,” *J. Syst. Eng.*, **1**(4), pp. 267–284.
- [6] Jimenez, B., and Asano, T., 2008, *Water Reuse: An International Survey of Current Practice, Issues and Needs*, IWA, London.
- [7] Dori, D., 2002, *Object-Process Methodology*, Springer-Verlag, Berlin.
- [8] Steward, V., 1981, “The Design Structure System: A Method for Managing the Design of Complex Systems,” *IEEE Trans. Eng. Manage.*, **28**, pp. 71–74.
- [9] Danilovic, M., and Browning, T., 2007, “Managing Complex Product Development Projects With Design Structure Matrices and Domain Mapping Matrices,” *Int. J. Proj. Manage.*, **25**(3), pp. 300–314.
- [10] Alexander, C., 1964, *Notes on the Synthesis of Form*, Harvard University Press, Cambridge, MA.
- [11] Suh, N., Bell, A., and Gossard, D. 1978, “On an Axiomatic Approach to Manufacturing and Manufacturing Systems,” *ASME J. Eng. Ind.*, **100**, pp. 127–130.
- [12] Eggert, R. J., 2004, *Engineering Design*, Prentice-Hall, Upper Saddle River, NJ.
- [13] Crawley, E., 2003, *Lecture Notes for ESD.34 System Architecture*, MIT, Cambridge, MA.
- [14] Buede, D., 2009, *The Engineering Design of Systems: Models And Methods*, Wiley, Hoboken, NJ.
- [15] The Webster Online Dictionary, <http://www.webstersonline.com/>
- [16] Eppinger, S., Nukala, M., and Whitney, D. 1997, “Generalized Models of Design Iteration Using Signal Flow Graphs,” *Res. Eng. Des.*, **9**, pp. 112–123.
- [17] Eppinger, S., 1991, “Model-Based Approaches to Managing Concurrent Engi-

- neering," J. Eng. Design, 2(4), pp. 283–290.
- [18] Baldwin, C. Y., and K. B. Clark 2000, *Design Rules Vol. 1: The Power of Modularity*, MIT, Cambridge, MA.
- [19] Warfield, J., 1973, "Binary Matrices in Systems Modeling," IEEE Trans. Syst. Man Cybern., **SMC-3**(5), pp. 441–449.
- [20] Sharman, D., and Yassine, A., 2004, "Characterizing Complex Product Architectures," J. Syst. Eng., 7(1), pp. 35–60.
- [21] McCormack, A., Rusnak, J., and Baldwin, C., 2006, "Exploring the Structure of Complex Software Designs: An Empirical Study of Open Source and Proprietary Code," Manage. Sci., **52**(7), pp. 1015–1030.
- [22] Foster and Partners Abu Dhabi Future Energy Company Masdar Development Project, 2007, Detailed Masterplan, Master Plan, Vols. 1–4.
- [23] Locke, S., Middle East Business Intelligence (MEED).
- [24] Cho, S.-H., 2001, "An Integrated Method for Managing Complex Engineering Projects Using the Design Structure Matrix and Advanced Simulation," S.M. thesis, MIT, Cambridge, MA.
- [25] Eppinger, S., and Gebala, D., 1991, "Methods for Analyzing Design Procedures," ASME Conference on Design Theory and Methodology, Miami, FL, Sept., pp. 227–233.