9TH INTERNATIONAL DESIGN STRUCTURE MATRIX CONFERENCE, DSM'07
16 – 18 OCTOBER 2007, MUNICH, GERMANY

THE PROJECTION RELATIONSHIP BETWEEN OBJECT PROCESS MODELS (OPM) AND DESIGN STRUCTURE MATRICES (DSM)

Edward Crawley[1], Justin Colson[2]

[1]Professor of Aeronautics and Astronautics and Professor of Engineering Systems, Massachusetts Institute of Technology
[2]Graduate Research Assistant, Massachusetts Institute of Technology

1 INTRODUCTION

Object Process Models (OPM) [1] provide a bipartite graphical means to represent a very general complex system of operand objects, processes and instrument objects, but to date do not have a complementary matrix representation that would facilitate computation. Design Structure Matrices [5], [6] provide more compact matrix representations of interactions in a system, which facilitates computation, but only represent a restricted class of system relationships. In many ways, DSMs represent a projection into a subspace of a limited class of relationships/topologies of more general bipartite OPM graphs/systems [2], [3], [4]. In this paper, we define many of the similarities between these two types of system representations, and derive a general algorithm that projects any arbitrary system from an Object Process Model to a Design Structure Matrix. This facilitates both compact matrix representations and potentially computation. In the process, the assumptions and limitations of a DSM representation are also discussed.

2 BACKGROUND

Object Process Diagrams (OPDs) can be used to explicitly represent all causal relationships within a system. This system representation is built up of objects and processes. *Objects* are things that exist or have the potential for existence, and have states. *Processes* are transformations that can change the states of objects. Objects are further divided into *instrument objects* and *operands*, where the distinction is that the instrument is the agent of the process, while the operand is the object whose states are affected by the process. An OPD explicitly calls out the relationship between the objects and processes in a system with semantically exact links or edges. For example, a simple and generic OPD might look like:

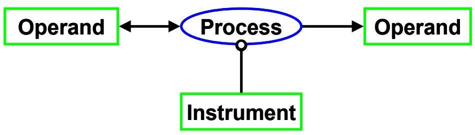


Figure 1. Simple Object Process Diagram

Where each rectangle denotes an objects and each oval denotes a process. This OPD would be read as indicating the bottom instrument is an agent of the process (round headed arrow), the left operand is effected by the process (double headed arrow), and the right operand is created by the process (single headed arrow leading away from the process). One operand, one process, and one instrument object is the canonical structure of a system (as it is a sentence in natural human language), and all complete descriptions of systems must have a means of representing these elements and their interrelationships.

Design Structure Matrices implicitly show the causal relationships between different system components of interest. The flexible matrix representation is compact, and facilitates representation, simple communications with others, and potentially computation. There are limitations to DSM representations, some of which are operational, and some more fundamental. Operationally, a specific DSM is usually created to represent a single type of causal relationship, such as the relationship of processes with other processes. Objects, both instruments and operands, are ignored or implicit. Alternatively, DSMs sometimes represent the objects (operands and/or instruments) on the sides of the N squared diagram, making the processes implicit. Sometimes a less precise user of DSMs will mix the objects and processes. A second operational limitation is the description of boundary interactions. If the DSM is constructed of the elements of the system, there is no representation of interactions across the boundary of the system. A more fundamental limitation of the DSM is its difficulty in representing relationships that are not binary, connecting two elements at a time, but fundamentally link three or more elements. A second fundamental difficulty is encountered where there is not a one to one relationship between processes and instrument objects. Both of these fundamental challenges of DSMs are easily resolved in the bipartite graph representation of an OPD.

3 THE PROJECTION RELATION

In this paper, we attempt to bridge the gap between these two very different system representations, producing a way of translating from OPD to DSM representations. The result is that we are able to use Object Process Diagrams to fully outline a unified system model, which we can then translate into a Design Structure Matrix representation. This unified model, presented in DSM form, can then be manipulated in a fashion only possible in the matrix representation.

The paper will detail the specifics involved in developing a general algorithm that projects any arbitrary Object Process Diagram system into a Design Structure Matrix. We show that it is possible to construct a matrix representation of the OPD that is N squared and causal. This matrix representation can then be partitioned into the visible and hidden modes. These visible modes represent the modes that have causal relationships (including feedback loops) on other interesting aspects of the system. Next, we construct and utilize a projection operator that projects the system into the M x M space represented by the visible system modes. Finally, a relationship is derived in which to project this arbitrary M x M Object Process Model into a Design Structure Matrix.

The main steps of deriving and using the projection relation are:

- An OPD of the system is developed, showing the system intended for analysis and immediately adjoining elements.
- The boundary of the system for analysis is defined, identifying external interfaces and elements, that are then taken as exogenous to the system under study.
- Each element (objects and process) of the system under study is isolated, and the topology of the system encoded in an "equation" of objects, processes and linkages – the internal elements are homogeneous terms, while the external exogenous elements are non-homogeneous terms.
- This set of equations can be represented by a matrix of objects and processes.
- The elements which will be represented in the DSM are selected (the visible elements).
- Matrix like manipulation, as well as the definition of a network projection inverse, allow the representation of the DSM like view based on the projection of the OPD view.

In the paper, a more complete mathematical representation of this process, and its derivation, will be presented.

4 OPM/DSM SIMILARITY PRINCIPLES

In the course of deriving and exercising the projection relation between OPDs and DSMs, a number of observations on the similarity of the two representations, and their relative limitations, can be made.

In general, DSMs are extremely useful in representing a very specific type of causal relationship, in which a directional flow of some sort passes from element to element. The developer of a DSM implicitly must make a decision as to which of the three canonical elements, the operands, processes or instruments are to be represented on the sides of the N^2 matrix. A relatively rare alternative is to include operand causal relations, where the flow being passed is, for example, a relationship among the operands. A common choice for the elements to be represented in the DSM are the processes, especially when modelling technical or business tasks. In this case, the entry of a mark into the causal relationship box in the DSM usually implicitly represents an operand being passed between the two processes. However, the presence or nature of the operand is hidden. Likewise, the instruments responsible for the process are not even implicitly represented.

An alternative DSM model can be formed by selecting the instrument objects to place along the sides of the N squared diagram. The entry into the causal relationship box is still often the operand passed by the processes associated with the instruments. The information about the processes is implicit. A complete representation of the operands, processes, and instruments would potentially require several DSMs (much like SysML requires several views). Although the resulting representation of the system is accurate and computable, some information can be lost along the way.

OPDs are much more general in nature than DSMs. They outline all of the causal and non-causal relationships within the system of interest, and are generally clear in examining multiple different relationships in a single diagram. Although complete and accurate, OPDs are often very complex in nature. Also, due to the generality of the system representation, with a focus on completeness of relationships, it is difficult to develop a computational means of analyzing the system [2], [3], [4].

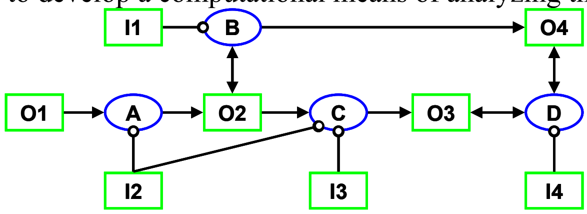


Figure 2. Generic OPD model

Process

	A	B	C	D
A				
B	O2			
C	O2	O2		
D		O4	O3	

Operand

	O1	O2	O3	O4
O1				
O2	A			
O3		C		D
O4		B	D	

Figure 3. Two DSM representations of the generic OPD model

The limitations of a DSM representation begin to be revealed by this projection of the more general OPD onto a DSM. For example, there are many relationships in a system that are not causal in nature. The relationship between an instrument and process is an example. This relationship is a steady fact, and does not imply anything happens first or second. In addition, there are, in general, more complex topologies of system than simply ones where a flow of some type leaves one element and enters another. Conditional information is an example of this kind of relationship. A third limitation of DSMs is their inability to capture relationships between processes that arise other than through passed operands. An example can be found in shared instruments. Finally, there is an ambiguity in DSMs that arises when more than two operands are linked by a process.
The details of these limitations will be elaborated upon, and examples presented in the paper.

REFERENCES

[1] Dori D., *Object-Process Methodology: A Holistic System Paradigm*, 2002, Springer-Verlag Berlin/Heidelberg.

[2] Koo B. H. Y., A Meta-language for Systems Architecting. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2005.

[3] Koo B. H. Y., Simmons W. L. and Crawley E. F., Algebra of systems: an executable framework for model synthesis and evaluation. In Proceedings of the *2007 International Conference on Systems Engineering and Modeling*, 2007.

[4] Simmons W. L., Koo B. H. Y. and Crawley E. F., A Computational Method for Mapping the Decision Space of the Lunar Exploration Program, Proceedings of *57th International Astronautical Congress 2006*, Valencia, Spain, October 2006, IAC-06-D1.3.7.

[5] Steward D. V., The Design Structure System: A Method for Managing the Design of Complex Systems, *IEEE Transactions on Engineering Management, vol. 28*, pp. 71-74, 1981.

[6] Black T. A., Fine C. H., and Sachs E. M., A Method for Systems Design Using Precedence Relationships: An Application to Automotive Brake Systems, M.I.T. Sloan School of Management, Cambridge, MA, Working Paper no. 3208, 1990.

Contact: Edward Crawley
Massachusetts Institute of Technology
Aeronautics and Astronautics
77 Massachusetts Avenue
Cambridge, MA 02139-4307
USA
Phone: +1 617 253 7510
Fax: +1 617 324 6901
e-mail: crawley@mit.edu

9TH INTERNATIONAL DSM CONFERENCE

The Projection Relationship Between Object Process Models (OPM) and Design Structure Matrices (DSM)

Edward Crawley, Justin Colson, Maokai Lin

Engineering Systems Division
Massachusetts Institute of Technology

IN COOPERATION WITH BMW GROUP
CAPITALIZE ON COMPLEXITY

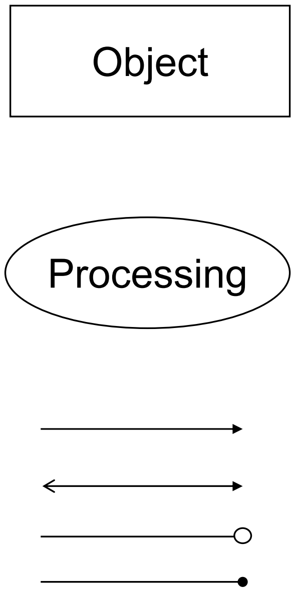
Index of Topics

- DSM's - compact and "computable" matrix models
  - Represent directed graphs with pairwise connections
- OPM's - bipartite graph models
  - Objects and process, directed and directionless, multi-element connections
  - More complex, but more information
- A projection relationship exists between OPM's (for systems representable by DSM's) and their DSM representation
- Projection clarifies/reveals:
  - The information actually stored in the DSM
  - The generalized DSM of systems not conventionally represented by DSM's

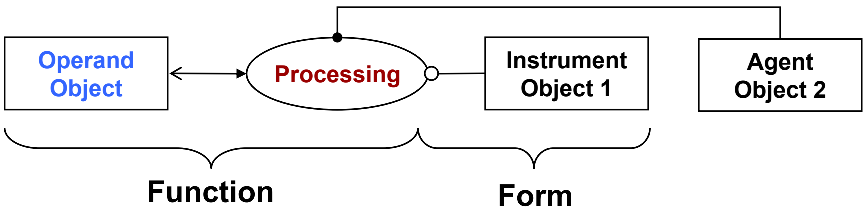
Object Process Modeling

- Object: that which has the potential of stable, unconditional existence for some positive duration of time.
- Process: the pattern of transformation applied to one or more objects. Processes change an object's state.
- All links between objects and processes have precise semantics:
  - Directed input/output
  - Non-directed input/output - "effects"
  - Instrument (non-human)
  - Agent (human)

Ref: Object Process Methodology, Dov Dori, Springer, 2002

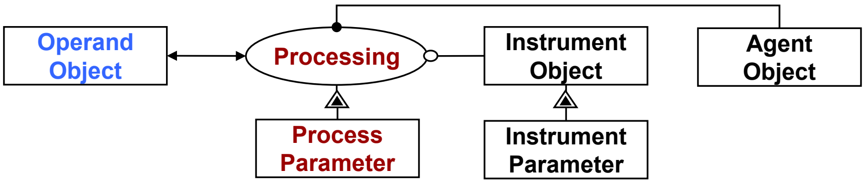


Canonical System Description



- System architecture of a system is made up of operands + processes (functions) plus instrument objects (form) and/or agents (objects)
- Formally read as: process effects operand, with object 1 as an instrument and object 2 as an agent, or less formally:
- Examples:
  - Heat is transferred with an evaporator
  - Report is prepared by the accounting team
  - Equipment is powered with an electric generating plant
  - Array is sorted with bublesort routine

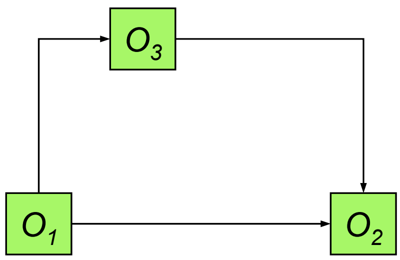
Relation of Canonical OPM with Types of DSM's



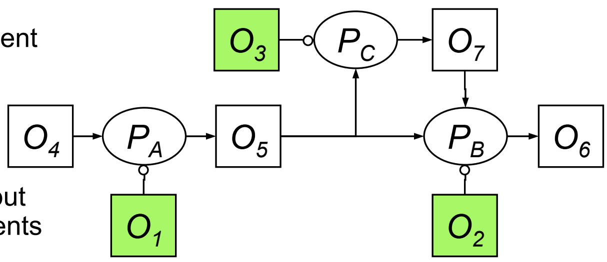
- Product/system/component DSM's are relationships among the **Instrument Objects**
  - But the process and operands are implicit or hinted at by the "transfer"
- Organizational/Team DSM's are relationships among the **Agent Objects**
  - But the process and operands are implicit or hinted at by the "transfer"
- Process/Activity DSM's are relationships among the **Processes and Operands**
  - Instruments and Agents are implicit
  - Activity DSM's are usually for organizational functions, but product functions also should be represented
- Product/system/component Parameter DSM's are relationships among the **Instrument Parameters and Process Parameters**
  - But the process and operands are implicit or hinted at by the "transfer"
- In a DSM, some of the canonical information is present, some implicit, and some absent, making the DSM likely be a **projection** of the full system representation

Case 1 - Component DSM to OPD of a Simple System

	1	2	3
Object 1			
Object 2	X		
Object 3	X	X	

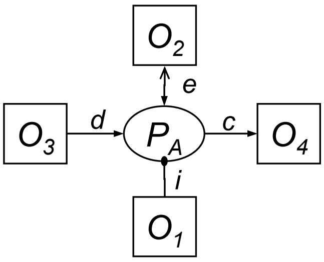


- Start with component DSM of the system that exhibits directed binary links
- The DSM shows directed links between components - instrument objects
- Components are actually instruments for processes - assume "independence"
- Operand objects are present, but must be inferred from components or the [smei] notation
- Note that boundary operands are very implicit in the DSM



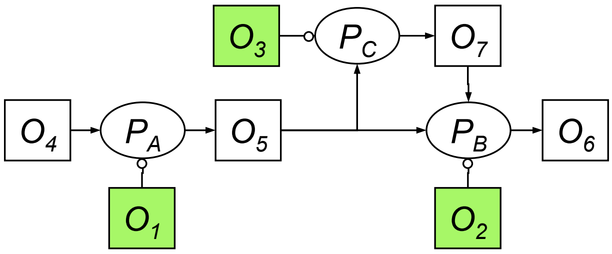
OPD for Case 1

Deriving Object Process Matrix (OPM) from OPD



	1	2	3	4
A	**i**	**e**	**d**	**c’**

- For nodes interior to the system, isolate each node and develop connectivity array for that node - if note is on “receiving end” of arrow, use x, if on sending end, use x’
- Assemble into Object Process Matrix
- Note diagonal nature of square blocks, transpose nature rectangular blocks



OPD and OPM for Case 1

	A	B	C	1	2	3	4	5	6	7
A	A			i			d	c'		
B		B			i			d	c'	d
C			C			i		d		c'
1	i'			1						
2		i'			2					
3			i'			3				
4	d'						4			
5	c	d'	d'					5		
6		c							6	
7		d'	c							7

“Sorting” OPM into Generalized DSM

	4	1	A	5	3	C	7	2	B	6
4	4		d'							
1		1	i'							
A	d	i	A	c'						
5			c	5		d'			d'	
3					3	i'				
C				d	i	C	c'			
7						c	7		d'	
2								2	i'	
B				d			d	i	B	c'
6									c	6

Sorted static generalized DSM

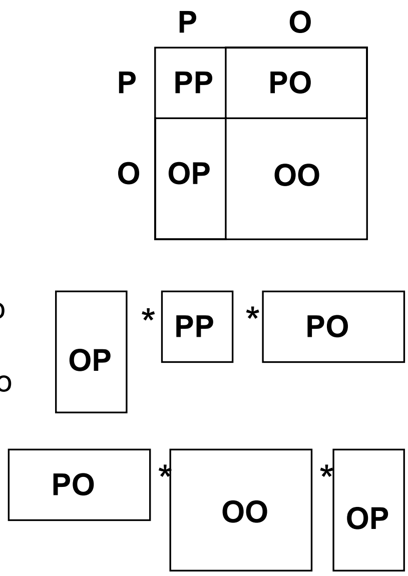
	4	1	A	5	3	C	7	2	B	6
4	4									
1		1								
A	d	i	A							
5			c	5						
3					3					
C				d	i	C				
7						c	7			
2								2		
B				d			d	i	B	
6									c	6

Sorted directed generalized DSM

- OPM can be sorted into static generalized DSM with both objects (instruments and operands) and processes on each side
- If only “forward”, “directed” or “causal” links are retained (the x ones) and the “reverse” or “anti-causal” ones dropped (the x’ ones), a directed generalized DSM is formed
- All DSM tools such as clustering, tearing, etc. can be applied to these matrices

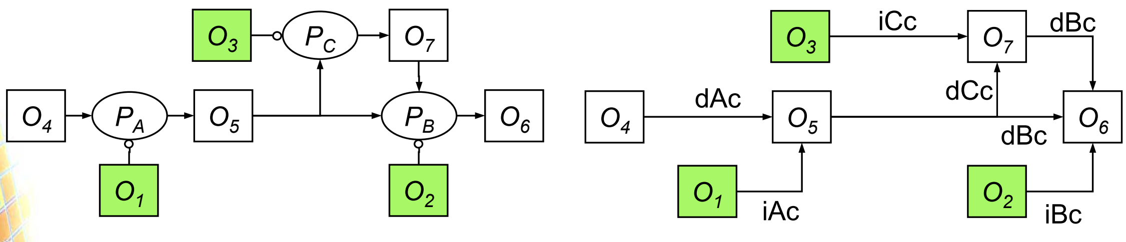
Creating Projected OPM Models

	A	B	C	1	2	3	4	5	6	7
A	A			i			d	c'		
B		B			i			d	c'	d
C			C			i		d		c'
1	i'			1						
2		i'			2					
3			i'			3				
4	d'						4			
5	c	d'	d'					5		
6		c							6	
7		d'	c							7



- A projection to a subspace of the model is done by first forming the Object Process Matrix
- Create the product of OP*PP*PO to project onto the object space
- Create the product of PO*OO*OP to project onto the process space
- Convert back to graph if desired
- Having projected onto the process or object spaces, further projection can be done in a similar manner to view a subset of the objects, for example, the components

Case 1 - Projection onto Objects - Component + Instrument DSM



Projected causal Object Diagram

$$\begin{bmatrix} i' & & \\ & i' & \\ & & i' \\ d' & & \\ c & d' & d' \\ & c & \\ & d' & c \end{bmatrix} * \begin{bmatrix} A & & \\ & B & \\ & & C \end{bmatrix} * \begin{bmatrix} i & & & d & c' & & \\ & i & & & d & c' & d \\ & & i & & d & & c' \end{bmatrix} =$$

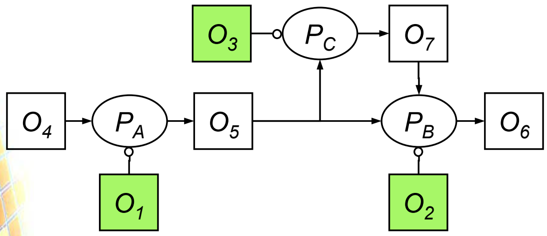
	1	2	3	4	5	6	7
1	i'Ai			i'Ad	i'Ac'		
2		i'Bi			i'Bd	i'Bc'	i'Bd
3			i'Ci		i'Cd		i'Cc'
4	d'Ai			d'Ad	d'Ac'		
5	cAi	d'Bi	d'Ci	cAd	cAc'+d'Bd+d'Cd	d'Bc'	d'Bd+d'Cc'
6		cBi			cBd	cBc'	cBd
7		d'Bi	cCi		d'Bd+cCd	d'Bc'	d'Bd+cCc'

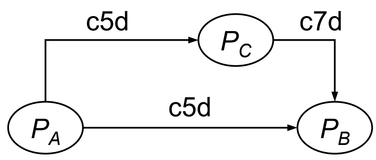
Examples:

O1->O5: O1 is an instrument of PA which creates O5

O5->O6: O5 is destroyed by PB which creates O6

Case 1 - Projection onto Processes - Process DSM





Projected causal Process Diagram

$$\begin{bmatrix} i & & & d & c' & & \\ & i & & & d & c' & d \\ & & i & & d & & c' \end{bmatrix} * \begin{bmatrix} 1 & & & & & & \\ & 2 & & & & & \\ & & 3 & & & & \\ & & & 4 & & & \\ & & & & 5 & & \\ & & & & & 6 & \\ & & & & & & 7 \end{bmatrix} * \begin{bmatrix} i' & & \\ & i' & \\ & & i' \\ d' & & \\ c & d' & d' \\ & c & \\ & d' & c \end{bmatrix} =$$

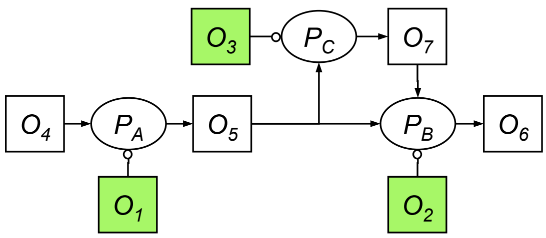
	A	B	C
A	i1i'+d4d'+c'5c	c'5d'	c'5d'
B	d5c	i2i'+d5d'+c'6c+d7d'	d5d'+d7c
C	d5c	d5d'+c'7d'	i3i'+d5d'+c'7c

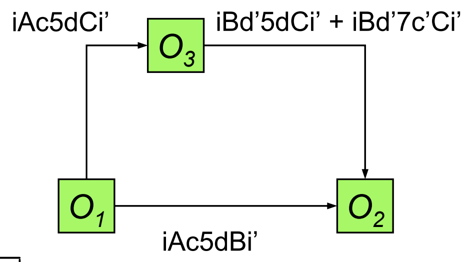
Examples:
PA->PB: PA creates O5 which is destroyed by PB

PC->PB: PC creates O7 which is destroyed by PB

IN COOPERATION WITH BMW GROUP
CAPITALIZE ON COMPLEXITY

Case 1 - Projection onto Instrument Objects Only





	1	2	3
1		i'Ac'5d'Bi	i'Ac'5d'Ci
2	i'Bd5cAi		i'Bd5d'Ci+i'Bd7cCi
3	i'Cd5cAi	i'Cd5d'Bi+i'Cc'7d'Bi	

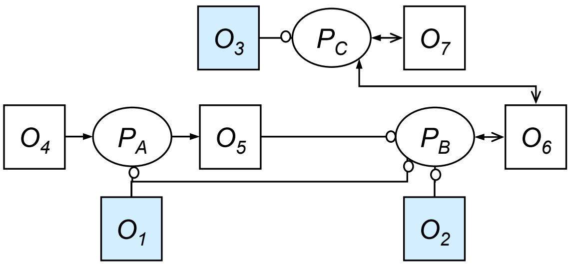
Examples:
O1->O2: O1 is an instrument of PA which creates O5 which is destroyed by PB with instrument O2

O2->O3: O2 is an instrument of PB which destroys O5 which is destroyed by PC with instrument O3; and O2 is and instrument of PB which destroys O7 which is created by PC with instrument of O3

- Instrument are linked through the processes they enable and the intermediate operand
- Note limitations in DSM: boundary operands absent, instrument links not strictly directed
- The projection of the OPD to instrument objects recovers the DSM with links containing exactly the information implicit in a component DSM

Case 2 - Non-directed and Non-binary Relations - Projection onto Objects

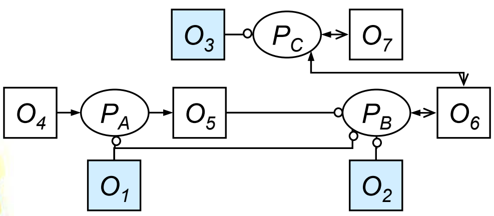
- Start with OPD of a system that exhibits non-directed and non-binary links
  - PC effects two operands
  - O5 was an operand of PA, and becomes an instrument of PB
  - O1, O2 and O5 are all instruments of PB (“independence” is violated)
- OPD and OPM show all of these factors explicitly, due to semantically exact graph notation, and bipartite nature of nodes



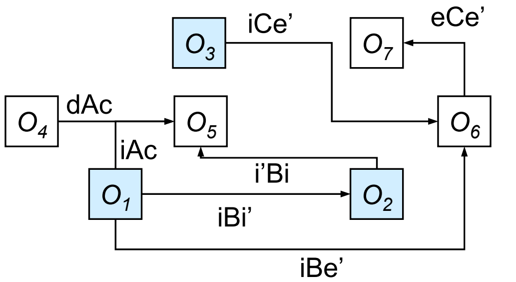
OPD and OPM for Case 2

	A	B	C	1	2	3	4	5	6	7
A	A			i			d	c'		
B		B		i	i			i	e	
C			C			i			e	e
1	i'	i'		1						
2		i'			2					
3			i'			3				
4	d'						4			
5	c	i'						5		
6		e'	e'						6	
7			e'							7

Case 2 - Non-directed and Non-binary Relations - Projection onto Objects



OPD for Case 2



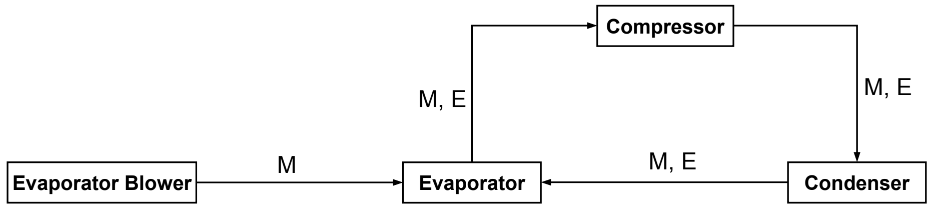
Projected Object Diagram (partial)

Projection onto Objects for Case 2

	1	2	3	4	5	6	7
1	i'Ai+i'Bi	i'Bi		i'Ad	i'Ac'+i'Bi	i'Be	
2	i'Bi	i'Bi			i'Bi	i'Be	
3			i'Ci			i'Ce	i'Ce
4	d'Ai			d'Ad	d'Ac'		
5	cAi+i'Bi	i'Bi		cAd	cAc'+i'Bi	i'Be	
6	e'Bi	e'Bi	e'Ci		e'Bi	e'Be+e'Ce	e'Ce
7			e'Ci			e'Ce	e'Ce

Note limitations in DSM:
- Unusual non-causal relations e.g. O1 to O6
- Unusual interrelationships e.g. O3, O7 and O6

Analyzing a Real Component DSM - Refrigerator DSM and Graph



- Well developed DSM contains:
  - Information on connections
  - Nature of connections
- But note: link between fan and evaporator only listed as mechanical - hard to represent the heat transfer process that occurs at the evaporator

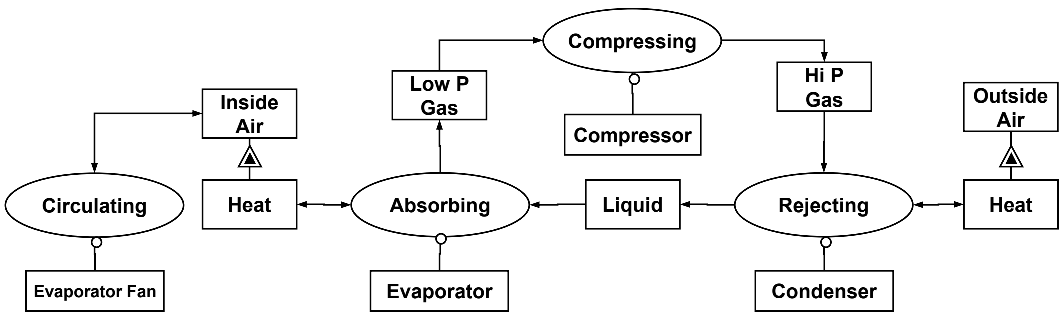
		A	B	C	D
Evaporator	A			0 2 0 2	0 0 0 2
Compressor	B	0 2 0 2			
Condensor	C		0 2 0 2		
Evaporator Blower	D				

Note: Spatial relations omitted example

Spatial	S	E	Energy
Information	I	M	Materials

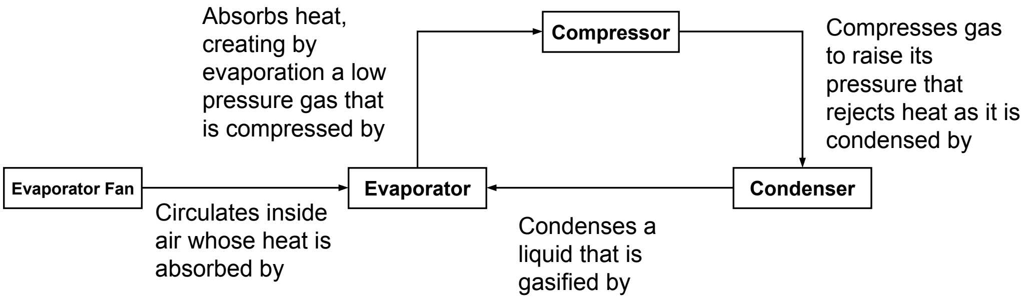
Ref: DSMweb.org and original work by Pimmler and Eppinger, 1994

OPD of Refrigerator



- OPD is much more explicit about Instrument Objects, Processes and Operands
- Able to explicitly show that the fan circulates the air (location state) while the evaporator draws heat from the air (thermal state)

Projection of the OPD to the Instrument Objects = Generalized DSM



- Projecting the OPD to the instrument object space creates a diagram similar to the DSM
- All of the information from the OPD is present, labeled along the directed links
- Boundary information such as interaction with outside air lost

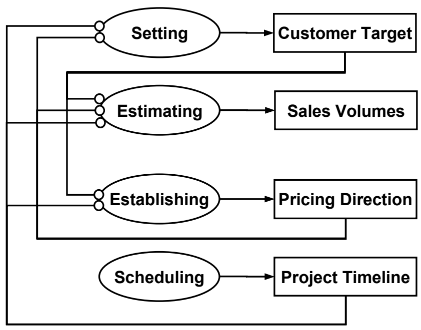
Semiconductor Development Process Example

		1	2	3	4
1	Set customer target	•		X	X
2	Estimate sales volumes	X	•	X	X
3	Establish pricing direction	X		•	X
4	Schedule project timeline				•

DSM

		Customer Target	Sales Volume	Pricing Direction	Project Timeline
1	Setting	*c*		*i*	*i*
2	Estimating	*i*	*c*	*i*	*i*
3	Establishing	*i*		*c*	*I*
4	Scheduling				*c*

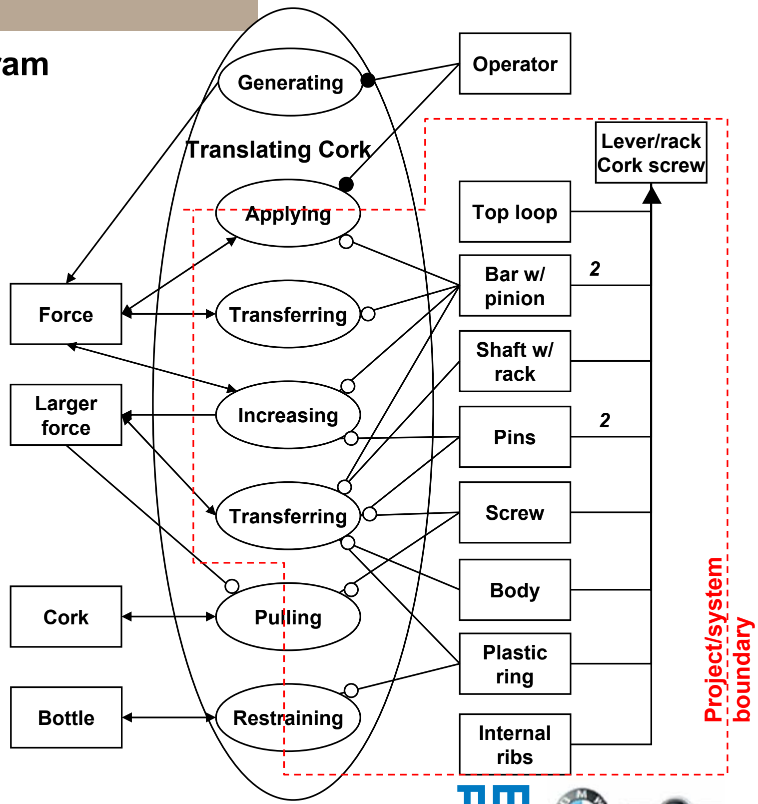
OPM



- In a development process DSM, the “tasks” are functions that contain a process and operand
- The process operand OPD and OPM reveal the same relationships as the DSM
- However, now other factors can be coupled to these processes, for example what team is involved
- Ref Eppinger - private communications

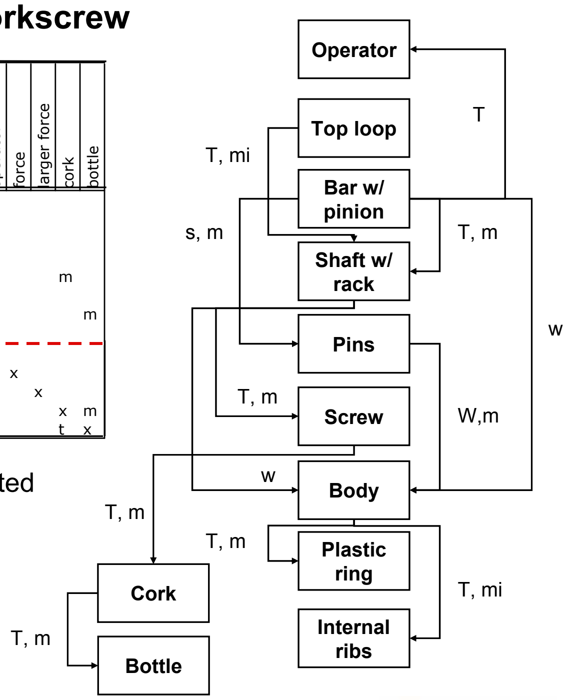
Object Process Diagram
Lever Corkscrew

- Full object process structure of the system
- Operands on left
- Processes in center
- Instruments and agents on the right
- No structural information of the objects alone



Object Object Diagram (Component DSM)
Lever Corkscrew

	top loop	bar with pinion	shaft with rack	pins	screw	body	plastic ring	internal ribs	operator	force	larger force	cork	bottle
top loop	x		mi										
bar with pinion		x	m	m					m				
shaft with rack	t	t	x		m								
pins		s		x		m							
screw			t		x							m	
body		w	w	w		x	m	mi					
plastic ring						t	x						m
internal ribs						t		x					
operator		t							x				
force										x			
larger force											x		
cork					t							x	m
bottle							t					t	x



- Only objects (components) represented
- Structural information of the formal (spatial) and implementation type
  - t touches
  - w within
  - s surrounds
  - m mechanically connected, etc.

Full Causal Object Process Matrix
Lever Corkscrew

	operator	generating	top loop	bar w/pinion	applying	transferring	pins	increasing	force	shaft w/rack	body	screw	plastic ring	transferring	larger frorce	pulling	internal ribs	restaining	cork	bottle
operator	x																			
generating	a	x																		
top loop			x																	
bar w/pinion				x																
applying	a			I	x															
transferring				I		x														
pins							x													
increasing				I			I	x	e											
force		c			e	e		e	x											
shaft w/rack										x										
body											x									
screw												x								
plastic ring													x							
transferring				I			I			I	I	I	I	x						
larger frorce								c						e	x					
pulling												I			I	x				
internal ribs																	x			
restaining													I					x		
cork																e			x	
bottle																		e		x

- Full object process structure of the system represented as an OPM
- No structural information of the objects alone
- Can be condensed to yield the Object projection, process projection, etc.

Summary

- DSMs and OPDs are both useful representations of systems, with their own strengths and limitations
- OPDs contain more explicit information on the system, and can represent more general relationships of elements, for example
  - Several object related through a process (rather than pairwise)
  - Objects related through agent and effecting (non-directed) relationships
  - Process that occurs at an instrument, rather than between
- A matrix representation, which captures all of the information present in an OPD, was developed, which facilitates "computation"
- A projection operator was demonstrated that condenses some information and leave others implicit. Various projections are possible:
  - Onto all objects (instruments and operands) [no DSM equivalent]
  - Onto instrument objects [equivalent to component or team DSM]
  - Onto process, or processes and operands [equivalent to activity DSM, but applicable to the product as well as the development process]
- When applied to the specialized classes of systems that can be modeled by DSMs, the condensed OPM is similar to the DSM, and reveals more explicitly the DSM