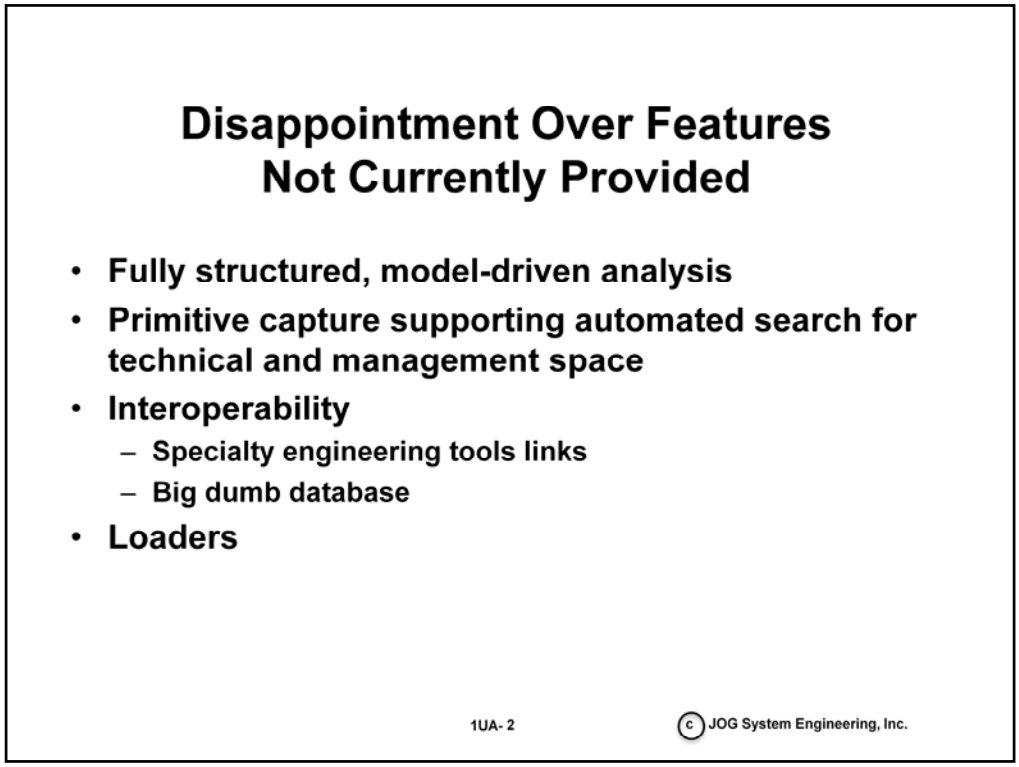
INCOSE San Diego Chapter
Annual Mini-Conference
October 25, 2008

What Happened to the Tool I Wanted?

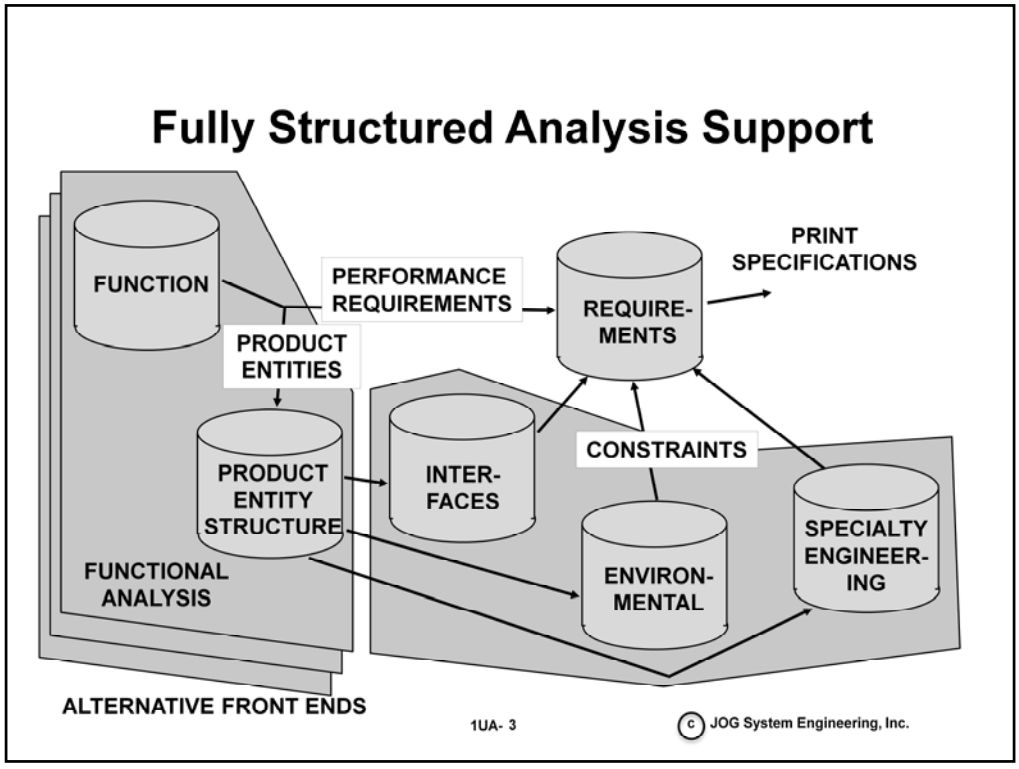
Jeffrey O. Grady
President JOG System Engineering
(858) 458-0121 jgrady@ucsd.edu

1UA-1

- **Years ago many of us had high expectations for effective database systems to support our requirements analysis and management needs.**
- **Many of us developed in-house systems because system engineering tools were slow in coming to market.**
- **Now many tools exist but they have missed some great opportunities and it is those opportunities that I would like to discuss today.**



- **There remain some features not yet supported.**
- **Tools do not capture primitive requirements that would permit capturing the quantity in a numerical field thus enabling some useful supporting mathematical treatment such as, "find me some management space." When we are in trouble we should be able to enlist the tool to review the content to locate where we might have excess capability of a similar kind to that which we are in trouble on. Perhaps the tool could tell us where we can re-allocate and get the extra capability we are looking for.**
- **The tools generally are isolated from the engineering models that are used by all of the different disciplines to do their jobs. The values in those source tools have to be transferred into the requirements database through new key strokes.**
- **Tools serving system engineering generally do not support modeling for the purpose of deriving design constraints.**



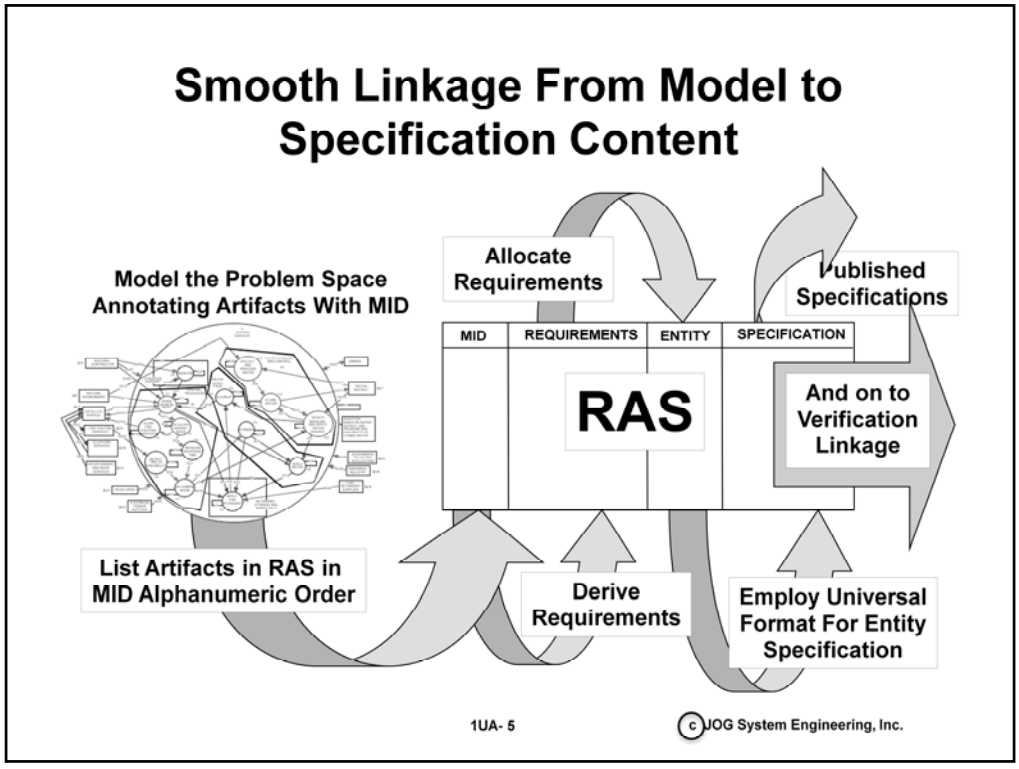
- **Few system engineering tools will support solution space modeling to the same degree they support problem space modeling.**
- **In particular environmental and specialty engineering are often not supported.**

RAS-Complete In Table Form

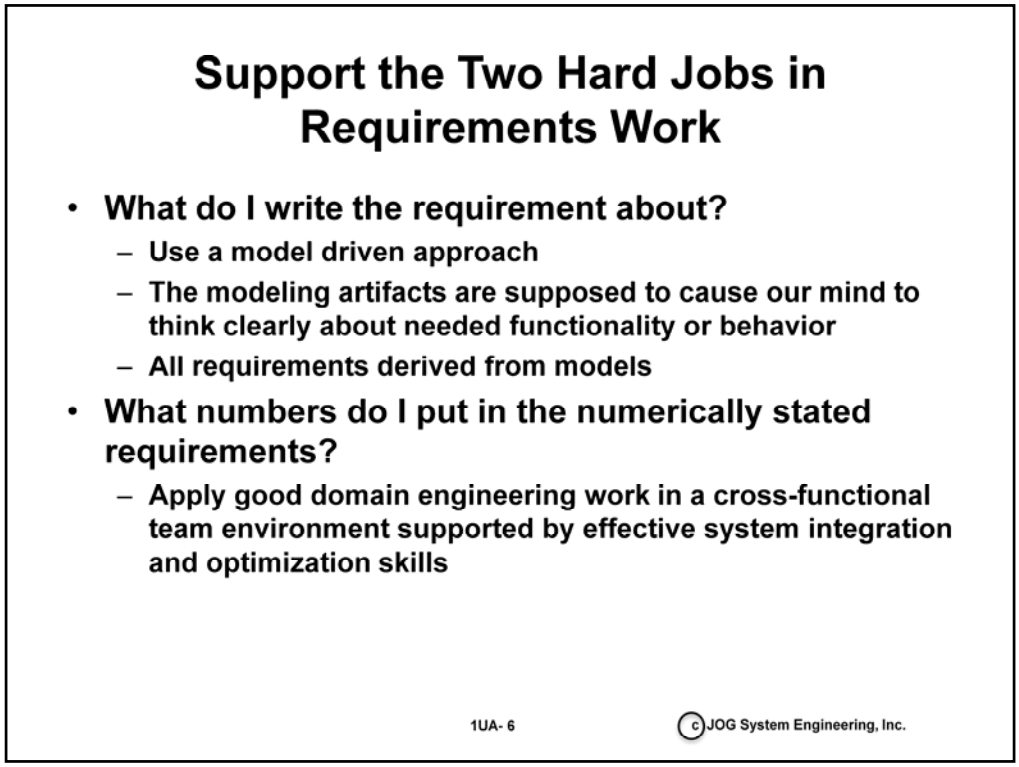
MODEL ENTITY MID	MODEL ENTITY NAME	REQUIREMENT ENTITY RID	REQUIREMENT	PRODUCT ENTITY PID	ITEM NAME	DOCUMENT ENTITY PARA	TITLE
F47	Use System			A	Product System		
F471	Deployment Ship Operations			A	Product System		
F4711	Store Array Operationally	XR67	Storage Volume < 10 ISO Vans	A1	Sensor Subsystem		
H	Specialty Engineering Disciplines			A	Product System		
H11	Reliability	EW34	Failure Rate < 10 x 10-6	A1	Sensor Subsystem	3.1.5	Reliability
H11	Reliability	RG31	Failure Rate < 3 x 10-6	A11	Cable	3.1.5	Reliability
H11	Reliability	FYH4	Failure Rate < 5 x 10-6	A12	Sensor Element	3.1.5	Reliability
H11	Reliability	G8R4	Failure Rate < 2 x 10-6	A13	Pressure Vessel	3.1.5	Reliability
H12	Maintainability	6GHU	Mean Time to Repair < 0.2 Hours	A1	Sensor Subsystem	3.1.6	Maintainability
H12	Maintainability	U9R4	Mean Time to Repair < 0.4 Hours	A11	Cable	3.1.6	Maintainability
H12	Maintainability	J897	Mean Time to Repair < 0.2 Hours	A12	Sensor Element	3.1.6	Maintainability
H12	Maintainability	9D7H	Mean Time to Repair < 0.1 Hours	A13	Pressure Vessel	3.1.6	Maintainability
I	System Interface			A	Product System		
I1	Internal Interface			A	Product System		
I11	Sensor Subsystem Innerface			A1			
I181	Aggregate Signal Feed Source Impedance	E37H	Aggregate Signal Feed Source Impedance= 52 ohms ± 2 ohms	A1	Sensor Subsystem		
I181	Aggregate Signal Feed Load Impedance	E37I	Aggregate Signal Feed Load Impedance= 52 ohms ± 2 ohms	A4	Analysis and Reporting Subsystem		
I2	System External Interface			A	Product System		
Q	System Environment			A	Product System		
QH	Hostile Environment			A	Product System		
QI	Self-Induced Environmental Stresses			A	Product System		
QN	Natural Environment			A	Product System		
QN1	Temperature	6D74	-40 degrees F< Temperature < +140 degrees F	A	Product System		
QX	Non-Cooperative Environmental Stresses			A	Product System		

1UA- 4

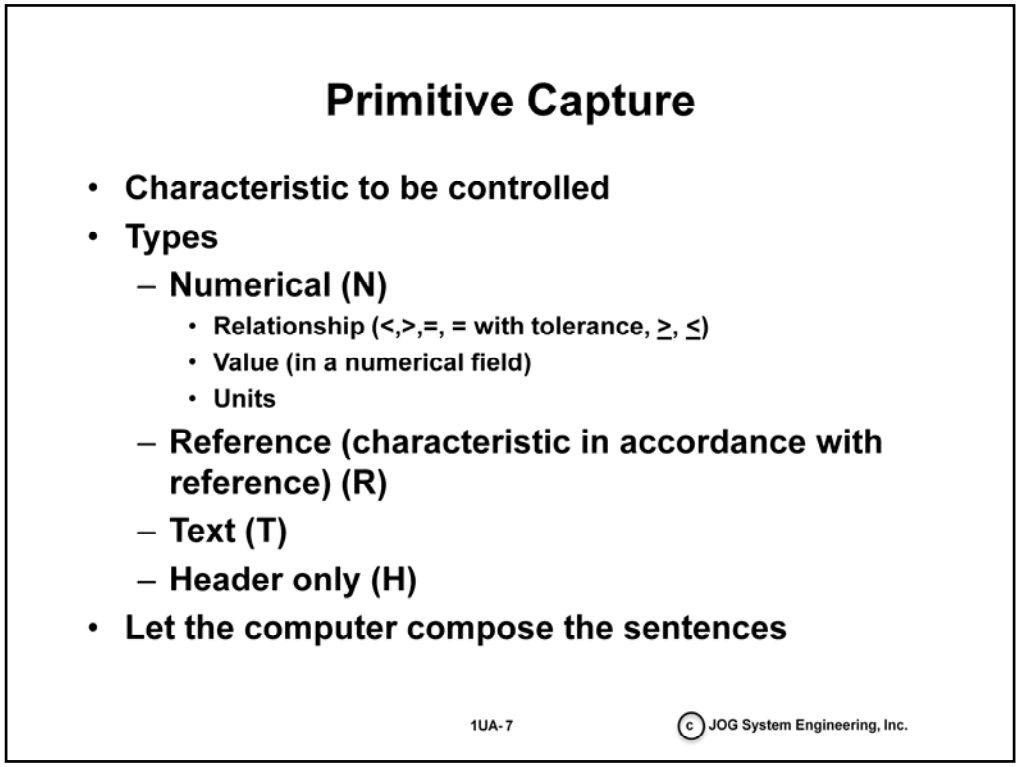
- **Our database should support a common RAS into which flows the modeling artifacts from which requirements are derived, the requirements so derived, the entities against which they collect, and finally the specification structures within which they are captured for however long we must prepare paper specifications.**
- **The author's method calls for identification of the modeling artifacts using a series of base 60 strings with leading letters that coordinate with specific modeling structures: F for functions in TSA, H for specialty engineering disciplines, I for interfaces, Q for environmental stresses.**
- **Other letters can be applied to the artifacts of other modeling methods.**



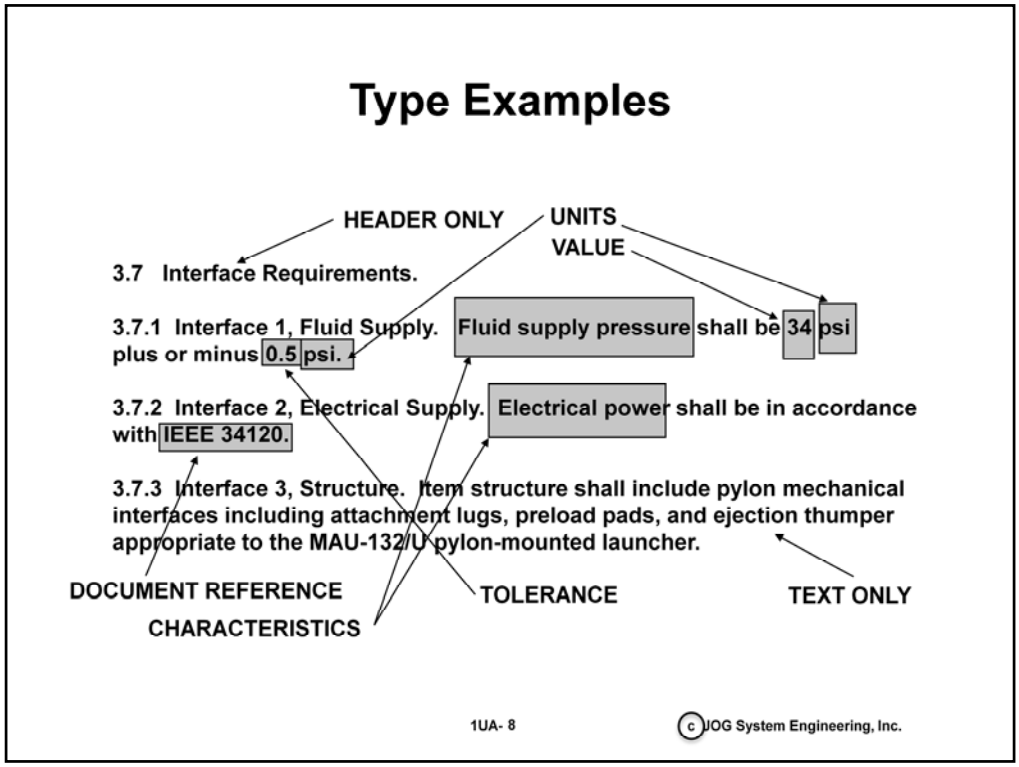
- **The database system should capture all of these entities and encourage human manipulation of them toward the goal of every requirement having been derived from a model.**



- **Some people think it is hard to phrase a proper sentence making a good requirement. That should not be our problem.**
- **The real difficulty comes in two forms: (1) what to write the requirements about and (2) what numbers to put in them.**
- **We solve the first problem by using models as the basis for requirements identification and good domain engineering to determine appropriate numbers.**



- **Primitive capture would require identification of a few different kinds of requirements statements because, unfortunately, all requirements are not numerically stated.**
- **Our tools should support the capture of the attributes to be controlled (derived from models) and the essential characteristics related to those attributes.**
- **In addition to numerical requirements statements we must also recognize header only, reference, and text formats.**
- **The computer can be made to form the requirement sentences that appear in the specification.**



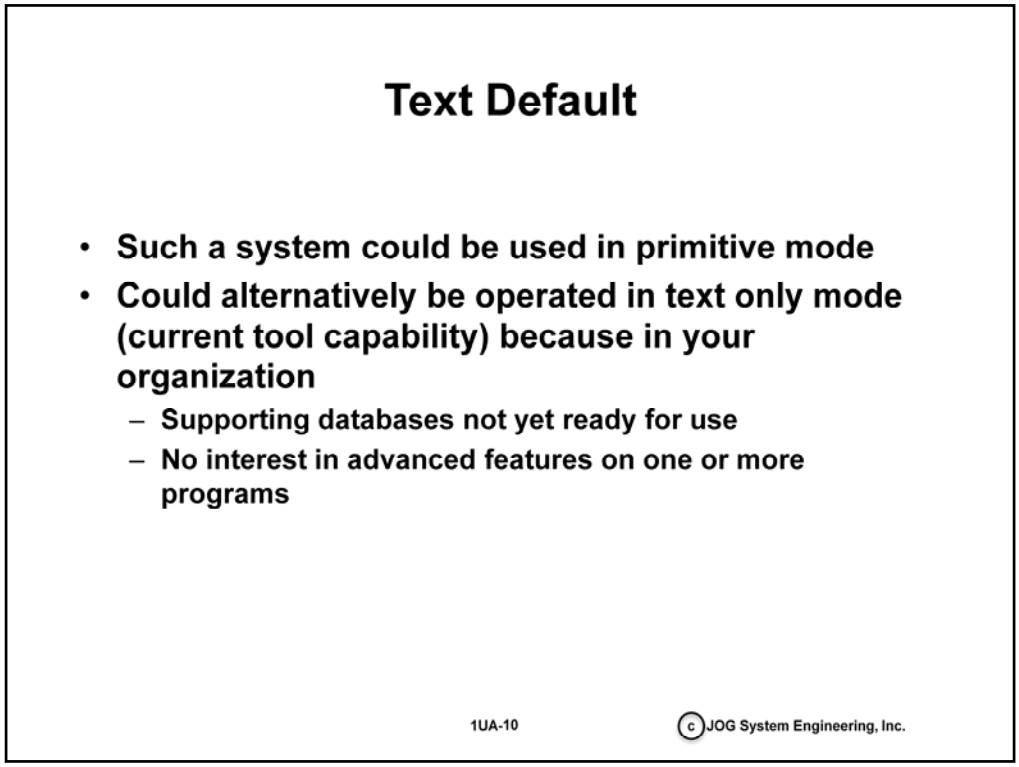
- **Here are four kinds of requirements statements listed on the prior chart. A system would do well to permit concatenation of text on the tail end of the reference or numerical type statements.**

Do Case Code to Print Using Primitive Capture

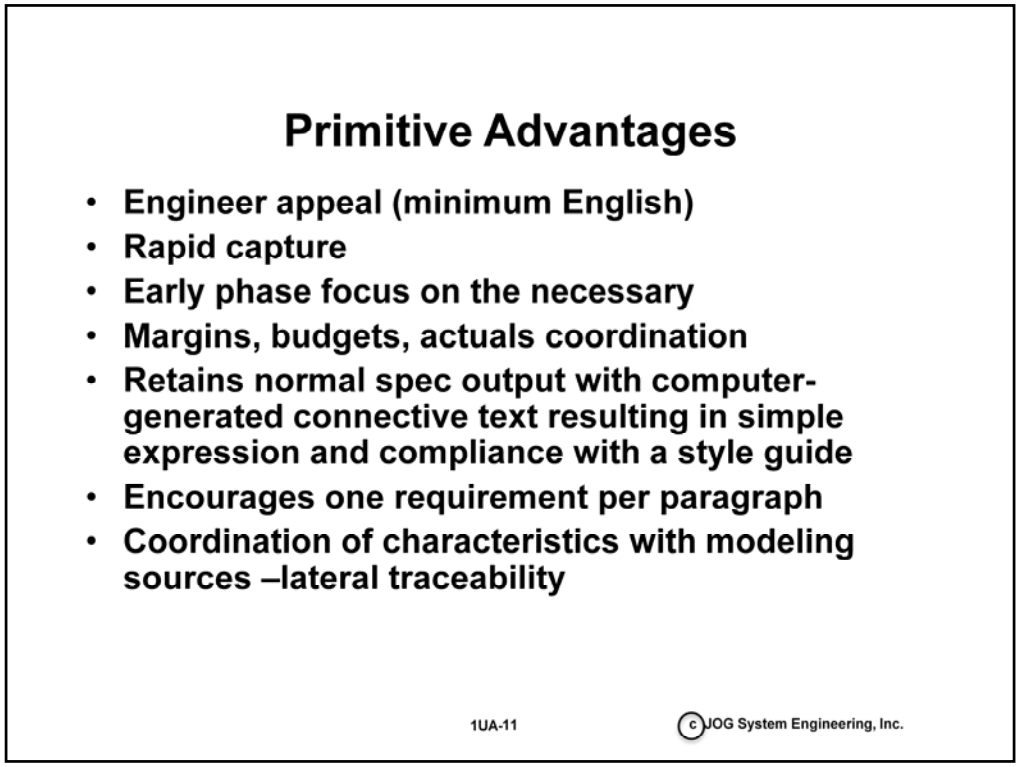
```
DOCASE
    CASE TYPE='H'
        @x,y SAY PARA+'   '+PARA_TITLE+'.'
    CASE TYPE='R'
        @x,y SAY PARA+'   '+PARA_TITLE+'.    Item '+TRIM(ATTRIBUTE)+'shall be in
        accordance with '+TRM(REFERENCE)+'.  '+TEXT
    CASE TYPE='T'
        @x,y SAY PARA+'   '+PARA_TITLE+'.    '+TEXT
    CASE TYPE='NL'
        @x,y SAY PARA+'   '+PARA_TITLE+'.   Item '+TRIM(ATTRIBUTE)+' shall be
        less than or equal to '+STR(VALUE)+' '+TRIM(UNITS)+. '+TEXT
            :
            :
ENDCASE
```

1UA- 9

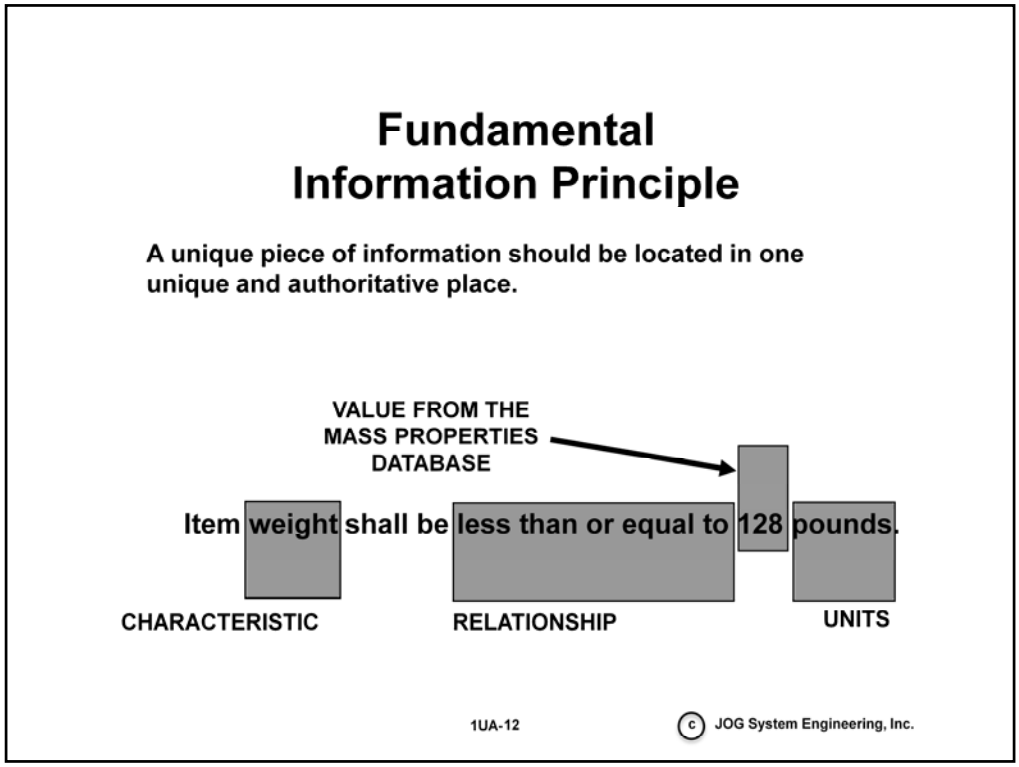
- **The system can easily be made to print the full specification language using simple case statements – here phrased in dBASE III+ format.**



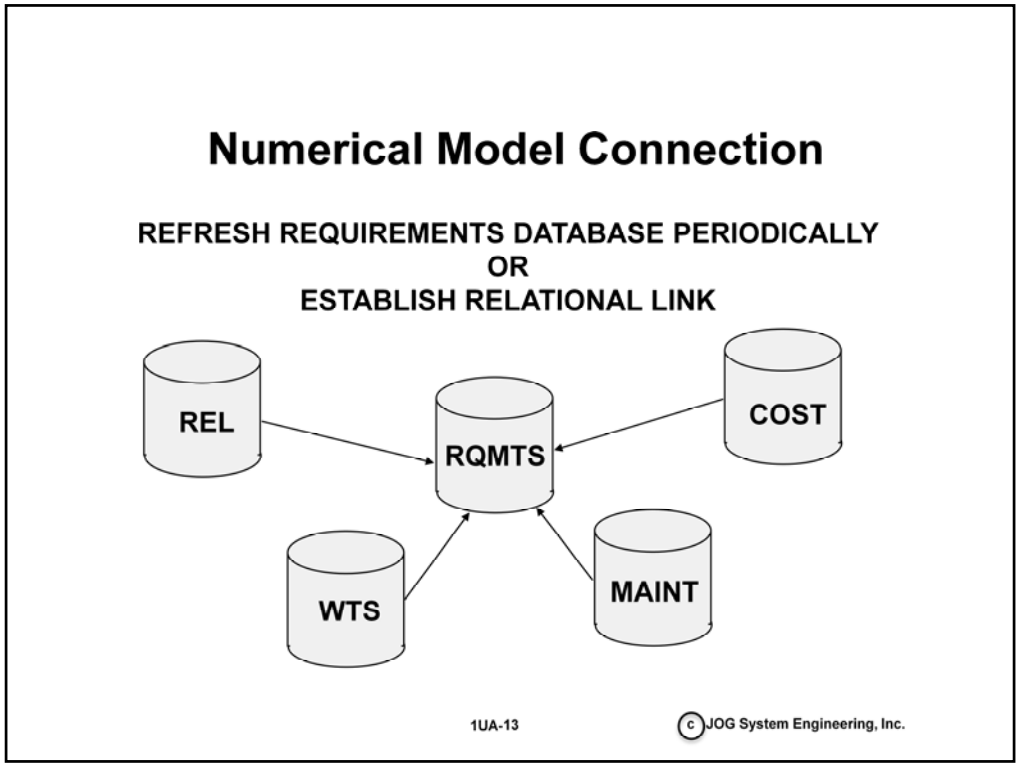
- **Such a tool could actually be used as a currently designed tool if one insisted.**
- **Every requirement would simply be a text type.**
- **This would be a big waste of capability but it could be done.**



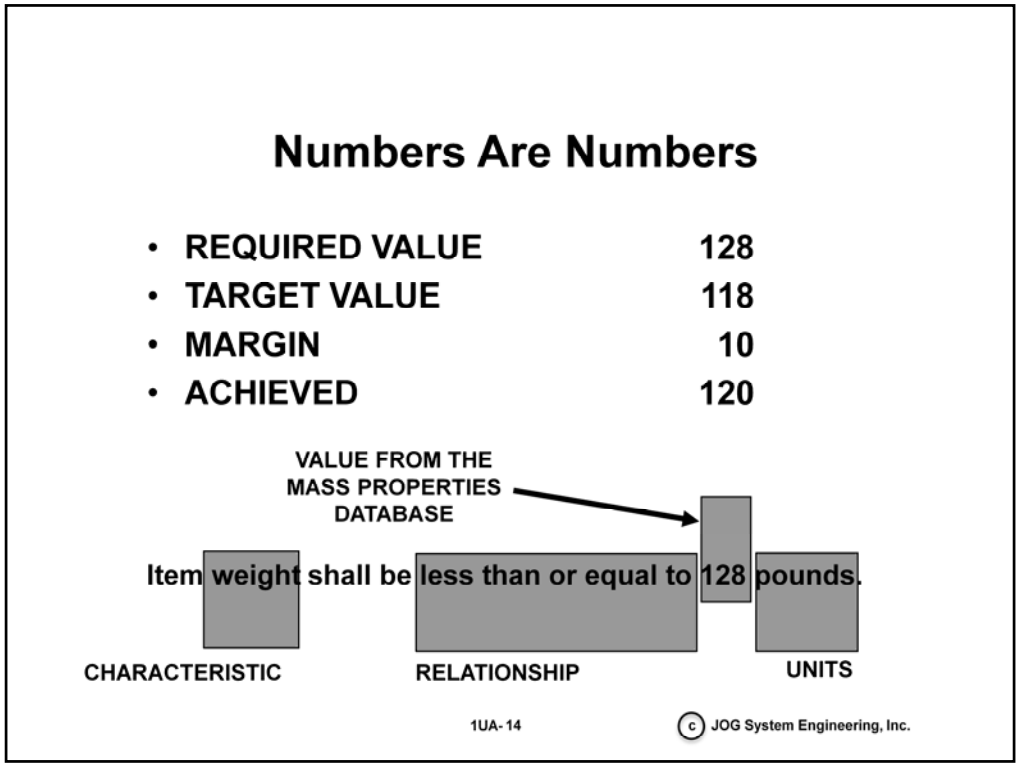
- **Such a system would have the advantage of freeing the system engineer from the burden of being able to craft proper English language sentences because the logic of the system would craft the sentences from the primitive components automatically. The other reasons listed are also supportive of good specification practices.**



- **Primitive capture combined with full coverage would preclude double booking of requirements. If the database system could simply go to the weights table database (maintained by mass properties people) and copy the weight figure for the item in question and insert it in the sentence it is forming for that paragraph, we could avoid the need for a system engineer to check the weights data base printout and enter the weights paragraph into the database with the weight figure contained in an ASCII string. We could avoid double booking of data.**



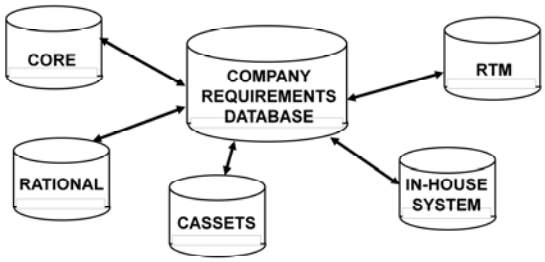
- **Ideally, the requirements database could either be refreshed at particular milestones from specialty engineering databases or linked as discussed on the prior chart.**



- **If our primitive system included the numerical value of the requirement in a numerical fields, it could provide some useful mathematical support relative to margins and budgets.**
- **We could tell the database to go find some management space when confronted by difficult problems.**

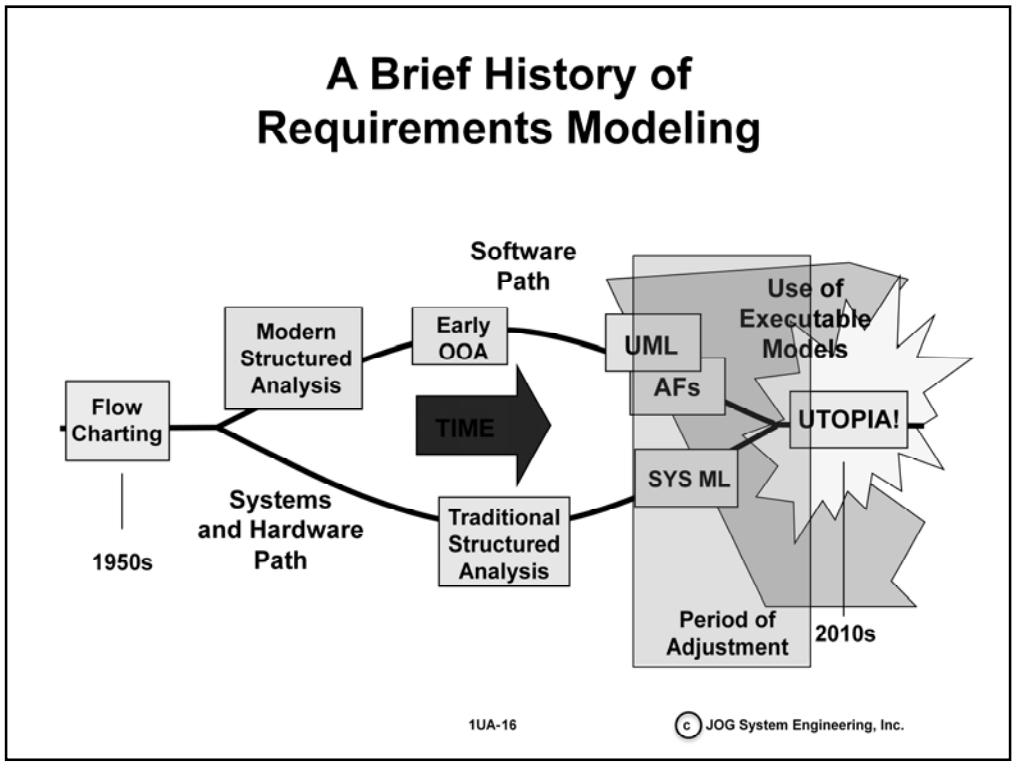
Interoperability

- Multiple tools may be used on a program
  - Hardware and software require separate tools today
- Provide traceability hooks across the tool gaps
- Requires standards and tool company cooperation against those standards

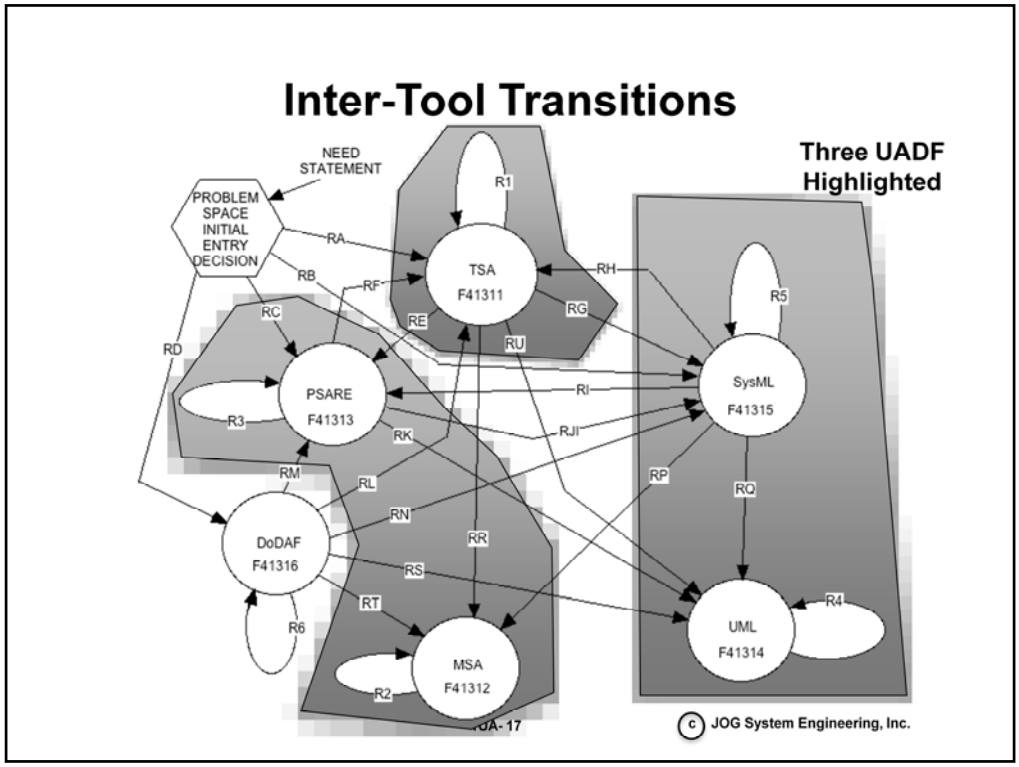


1UA-15

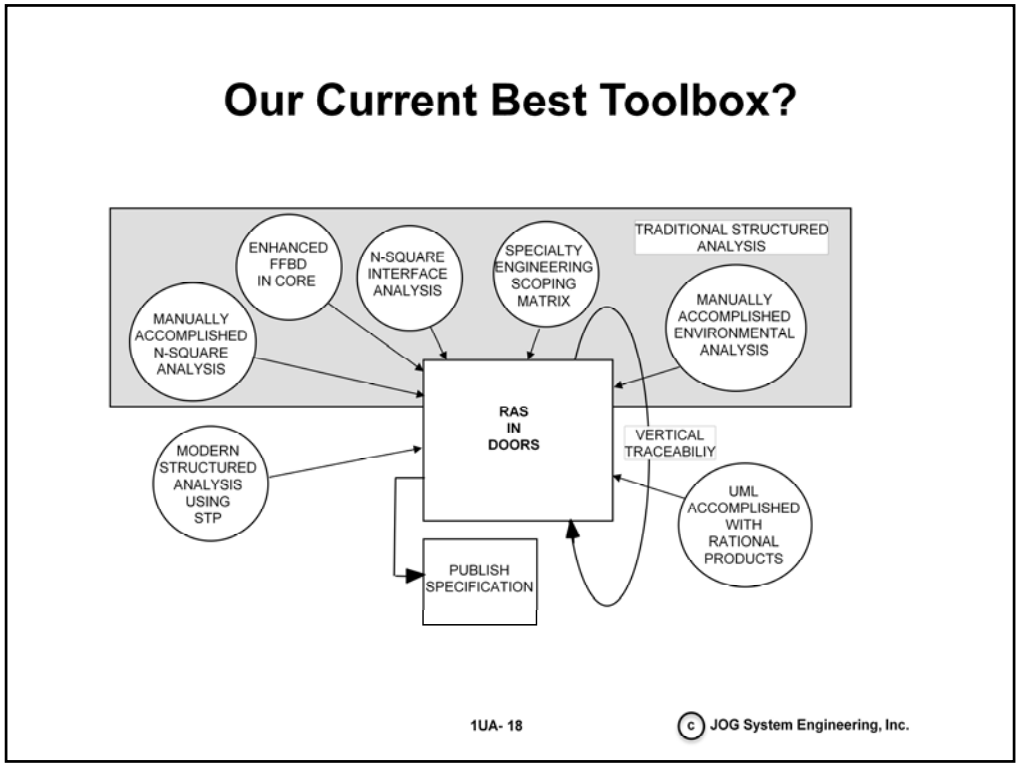
- **There does not seem to be any incentive for tool companies to cooperate in accordance with some agreed upon standards. Ideally, one could purchase whatever tools you preferred and they would be able to talk to each other. There is a lot of work going on in this area to define a language of data exchange and we will likely get to this point eventually.**
- **The lecturer prefers the use of what he calls a big dumb database that simply captures the text of the requirements statements and provides traceability support. The modeling in such a system could be done with pencil and paper with entry of the derived requirements into the big dumb database. Alternatively, one could use some mix of front end tools to model certain parts of the problem space and these requirements could be passed on to the big dumb database. Those kinds of links can be built and some companies employ them but ideally it would be easier than it is today.**



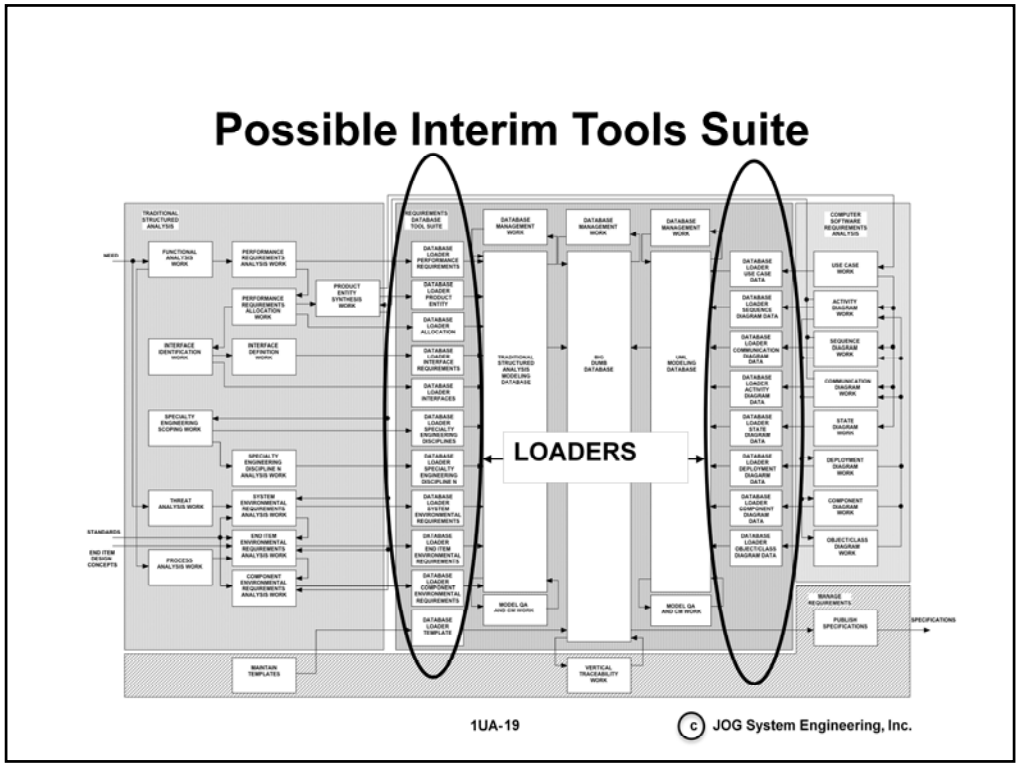
- **Over the past 40-50 years industry moved from a common development model for systems, hardware, and software in the form of flow charting to a branched approach where systems and hardware developers apply traditional structured analysis (derived from simple flow charting) and software developers apply one of a string of "new" techniques. The newest software technique, UML, promises to halt this split and rejoin systems, hardware, and software development in the same stream. Over time it is possible that UML and SysML will merge more tightly into a universal architecture description framework (UADF) modeling capability that will solve many of the difficulties experienced now in HW-SW integration.**



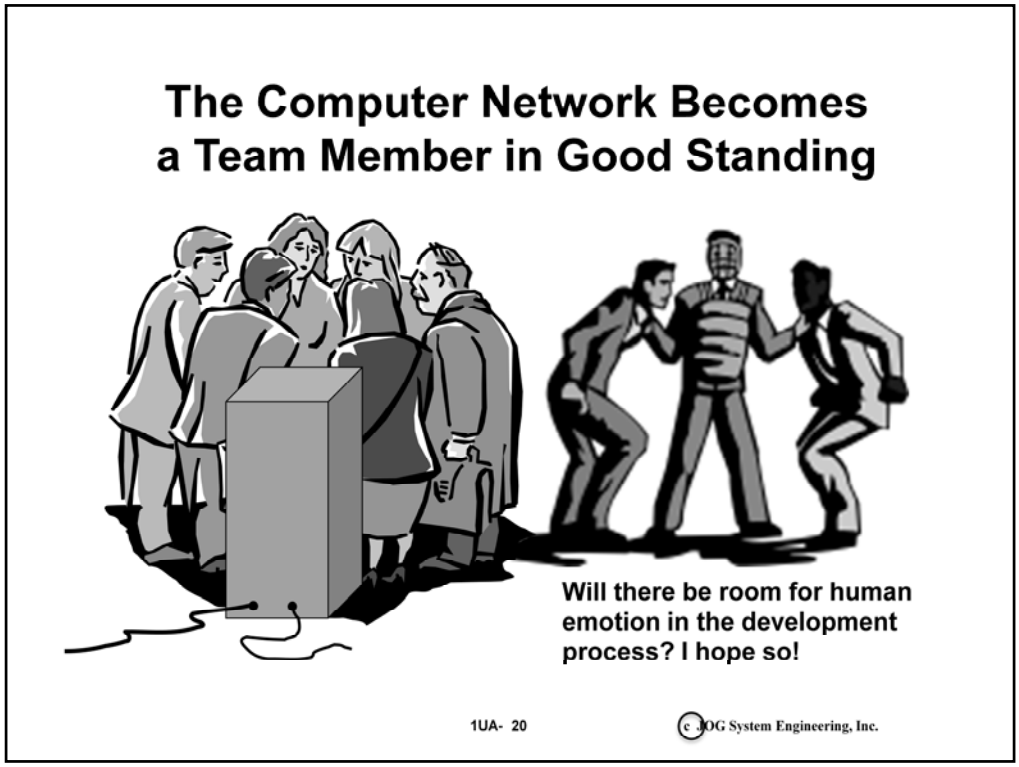
- **Three UDAF that can be formed to day are composed of functional analysis as performed in the 1960s across the whole product base, some combination of MSA and PSARE, or the combination of SysML and UML. Each of these combinations form a comprehensive set of models.**
- **Within any one of these UADF the number of possible inter-tool transitions is minimized.**



- **An enterprise may have some combination of tools it has purchased and are trying to apply on its programs in one or more combinations. DOORS may be used as the big dumb database.**



- **Today we might make a useful tool set with available tools by fashioning loaders in Excel that everyone will remain capable of using with the loaders transferring the data into up-to-date applications.**
- **Warren Smith, President of Execuspec, in Scottsdale, AZ has built some of these loaders.**



- **Teams are populated by many specialists who must communicate effectively to built the equivalent of one great mind attacking the problem space. They can be better aided by networked computer systems than currently is the case.**
- **Semi-automatic and automatic operation of model-driven development will employ the networked computers to evaluate the information on the islands created by specialists and run inter-model applications to spot and (in automatic mode) correct these errors.**