

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE PATENT TRIAL AND APPEAL BOARD

---

FORD MOTOR COMPANY  
Petitioner

v.

VERSATA DEVELOPMENT GROUP, INC.  
Patent Owner

---

Case IPR2016-01014  
Patent No. 5,825,651

---

**VERSATA'S PRELIMINARY RESPONSE  
UNDER 37 C.F.R. § 42.107(a)**

***Mail Stop "PATENT BOARD"***  
Patent Trial and Appeal Board  
U.S. Patent and Trademark Office  
P.O. Box 1450  
Alexandria, VA 22313-1450

## **TABLE OF CONTENTS**

TABLE OF AUTHORITIES .....	iii
EXHIBIT LIST .....	v
I. INTRODUCTION.....	1
II. BACKGROUND.....	3
A. Versata and Ford have a longstanding history.....	3
B. Versata developed the technology of the '651 patent to address the challenges of maintaining and configuring complex constraint-based systems. ....	5
C. Before the '651 patent, maintaining a configuration system required sophisticated programming knowledge to maintain complex, high-latency systems.....	7
D. To solve the challenge of maintaining and operating complex constraint-based systems, the '651 patent uses easy to maintain “product relationship(s)” and “part relationship(s)”. ....	9
E. Ford’s cited Axling reference describes a configuration system with the same maintenance complexities that the '651 patent was designed to overcome.	11
F. Ford’s cited Fohn reference likewise suffers from the same maintenance complexities that the '651 patent was designed to overcome. ....	18
G. The combination of Axling and Fohn represents nothing more than the complex, programmer-maintained constraint-based systems the '651 patent innovated upon. ....	21
III. CLAIM CONSTRUCTION .....	22
A. Relevant law .....	23
B. “product relationship(s)” .....	25
C. “part relationship(s)” .....	26
IV. ARGUMENT.....	27
A. Ford failed to show that the combination of Axling and Fohn teaches or suggests the claimed “product relationships.” .....	27
B. Ford failed to show that the combination of Axling and Fohn teaches or suggests the claimed “generating a set of part relationships between said one or more elements.” .....	35

C. The combination of Axling and Fohn fails to teach or suggest “said input is made without regard for the order” as recited in claim 71.....	40
D. Ford failed to show that Axling and Skovgaard are printed publications that qualify as prior art.....	41
1. Axling.....	43
2. Skovgaard.....	45
E. Ford’s petition lacks a proper obviousness analysis under <i>KSR</i> and <i>Graham</i> . 47	
1. Ford failed to articulate the differences between the applied art and the claimed subject matter.....	48
2. Ford fails to provide a reason to combine the applied references on an element-by-element basis.....	50
V. CONCLUSION.....	53
CERTIFICATION OF WORD COUNT (37 C.F.R. § 42.24(b)(1)) .....	54
CERTIFICATE OF SERVICE (37 C.F.R. § 42.6(e)) .....	55

## **TABLE OF AUTHORITIES**

### **Cases**

<i>Ariosa Diagnostics, Inc. v. Illumina, Inc.</i> , IPR2014-01093, Paper No. 69 (P.T.A.B. January 7, 2016) .....	42
<i>Black &amp; Decker, Inc. v. Positec USA, Inc.</i> , 2016 WL 2898012 (Fed. Cir. May 18, 2016).....	51
<i>Drinkware, LLC. v. National Graphics, Inc.</i> , 800 F.3d 1375 (Fed. Cir. 2015) .....	42
<i>Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.</i> , 800 F.3d 1375 (Fed. Cir. 2015) .....	27
<i>Graham v. John Deere Co.</i> , 383 U.S. 1 (1966) .....	47, 52
<i>In re Cronyn</i> , 890 F.2d 1158 (Fed. Cir. 1989) .....	46
<i>In re Klopfenstein</i> , 380 F. 3d 1345 (Fed. Cir. 2004) .....	46
<i>In re Magnum Oil Tools Int’l, Ltd.</i> , No. 2015-1300, 2016 WL 3974202 (Fed. Cir. July 25, 2016).....	27
<i>In re Wyer</i> , 655 F.2d 221 (C.C.P.A. 1981).....	42, 43
<i>KSR Int’l Co. v. Teleflex Inc.</i> , 550 U.S. 398 (2007) .....	41, 47, 51
<i>Liberty Mut. Ins. Co. v. Progressive Cas. Ins. Co.</i> , CBM2012-00003, Paper No. 7 (P.T.A.B. Oct. 25, 2012).....	50
<i>MIT v. AB Fortia</i> , 774 F.2d 1004 (Fed. Cir. 1985) .....	46

<i>Princeton Biochemicals, Inc. v. Beckman Coulter, Inc.</i> , 411 F.3d 1332 (Fed. Cir. 2005) .....	49
<i>Standard Innovation Corp. v. Lelo, Inc.</i> , IPR2014-00148, Paper No. 41 (P.T.A.B. April 23, 2015) .....	43
<i>Standard Innovation Corp.</i> , IPR2014-00148, Paper No. 41 .....	44
<i>Travelocity.com L.P. v. Cronos Techs. LLC</i> , CBM2014-00082, Paper No. 10 (P.T.A.B., Sept. 15, 2014) .....	41, 48, 49
<i>Trilogy v. Selectica</i> , 405 F. Supp.2d 731 (E.D. Tex. Dec. 20, 2005) .....	25
<i>United States v. Adams</i> , 383 U.S. 39 (1966) .....	47

## **Statutes**

35 U.S.C. § 102 .....	41
35 U.S.C. § 102(a) .....	45
35 U.S.C. § 102(b) .....	45
35 U.S.C. § 103 .....	48
35 U.S.C. § 103(a) .....	49
35 U.S.C. § 316(e) .....	42
35 U.S.C. § 102 .....	42

## **Regulations**

37 C.F.R. § 42.20(c) .....	50
----------------------------	----

**EXHIBIT LIST**

<b>Exhibit No.</b>	<b>Description</b>
<b>2001</b>	Complaint, <i>Versata Development Group, Inc. et al. v. Ford Motor Co.</i> , Civ. No. 4:15-cv-00316 (E.D. Tex.), D.I. 1.
<b>2002</b>	Versata: About Us, available at <a href="http://cpq.versata.com/aboutus">http://cpq.versata.com/aboutus</a> , 2016.
<b>2003</b>	“Ford Starts Firm to Manage Its Web Sites,” <i>ComputerWorld</i> , February 28, 2000.
<b>2004</b>	McCartney, Laton, “Trilogy Making A Name For Itself,” <i>ZDNet</i> , July 28, 2000.
<b>2005</b>	Field, Tom, “Suit Yourself,” <i>InsideCIO</i> , Vol. 10, No. 13, April 15, 1997.
<b>2006</b>	“Ford and Trilogy Launch Web Company,” <i>InformationWeek</i> , February 23, 2000.
<b>2007</b>	<i>Journal of Logic Programming</i> , Elsevier Science Inc., Vols.: 19 & 20, May & July 1994.
<b>2008</b>	“Versata Software Inc. Company Report,” Thomson Reuters MonitorSuite, May 2016.

## I. INTRODUCTION

Patent Owner Versata Development Group, Inc.<sup>1</sup> and Petitioner Ford Motor Company had a 19-year business relationship. During most of this time, Ford license Versata's patented Automotive Configuration Management (ACM) software and used this software as the backbone of its worldwide data infrastructure. (Ex. 2001, Complaint, pp. 1-2, in *Versata Development Group, Inc. et al. v. Ford Motor Co.*, (E.D. Tex. filed May 7, 2015).)

In 2011, to avoid paying licensing fees, Ford began a project to replace ACM with internally-developed software. Versata contends that this replacement software is a copy of ACM. (*See* Ex. 2001.)

Ford has filed an *inter partes* review petition seeking review of Claims 60-72 of U.S. Patent No. 5,825,651.<sup>2</sup> This patent protects some of the technologies used in ACM. Ford has requested *inter partes* review on three proposed grounds.

---

<sup>1</sup> Trilogy Software acquired Versata in 2006, at which point portions of Trilogy were merged with Versata as part of a reorganization. (Ex. 2008, Thomson report.) To avoid confusion, the Patent Owner will refer to both companies generically as Versata.

<sup>2</sup> The '651 patent is marked as Petitioner's exhibit 1001. Patent Owner will refer to this patent as "the '651 patent."

But Ford's petition is deficient. It fails to show that every claim element is found in the applied references, fails to establish that each of these applied references is prior art, fails to properly analyze obviousness. This deficient petition should be denied for at least five reasons.

*First*, Ford fails to show that the applied references teach or suggest the claimed "product relationships" of claim 60. However, the applied references describe a relationship between *classes*, rather than a product-to-*part* relationship.

*Second*, Ford fails to show that the applied references teach or suggest the claimed "part relationships" of claim 60. However, similar to the first deficiency, the applied references describe a relationship between *classes*, rather than a *part-to-part* relationship.

*Third*, Ford fails to show that the applied references teach or suggest the claimed "said input is made without regard for the order" required by dependent claim 71. Ford fails to clearly articulate how the purported ability of the applied references to "alter previously made configuration decisions" teaches or suggests that the input is made without regard for the order, rather than simply permitting an ability to backtrack and make changes to an earlier configuration decision in the order.

*Fourth*, Ford fails to show that the Axling and Skovgaard references are printed publications that qualify as prior art under 35 U.S.C. § 102, relying instead



on conclusory attorney arguments. But Ford has not come forward with any proof to show that Axling and Skovgaard were published. Without such proof, Ford has failed to show that Axling and Skovgaard are prior art.

*Finally*, Ford’s obviousness analysis throughout its petition is deficient. For instance, Ford provides generalized reasons for combining the Axling and Fohn references, but fails to explain why a person of ordinary skill in the art would have combined these references for each claim element. Additionally, Ford fails to identify any claim feature that is lacking from one of the references and supplied by the other—which is required for any proper obviousness analysis under *Graham*.

For the reasons summarized above and explained in more detail below, Ford has failed to establish a reasonable likelihood of prevailing on any of the proposed grounds of unpatentability. The Board should therefore deny institution this *inter partes* review proceeding against any of claims 60-72 of the ’651 patent.

## **II. BACKGROUND**

### **A. Versata and Ford have a longstanding history.**

Versata develops enterprise software that helps large corporations manage complex business operations. (Ex. 2002, Versata: About Us.) One of Versata’s most innovative areas of focus is product configuration software. (*See* Ex. 2003, “Ford Starts Firm to Manage Its Web Sites,” p. 2.) Although Versata has marketed

product configuration solutions in a variety of industries, this software has been particularly useful in the automotive industry. (Ex. 2004, “Trilogy Making a Name for Itself”); (Ex. 2005, “Suit Yourself.”)

Versata’s configuration technologies are particularly suited to automotive manufacturing because every vehicle is a complex system involving thousands of parts, features, and options. Complex software like Versata’s can enable automotive manufacturers to manage the billions of potential permutations that result from this complexity.

Between 1998 and 2014, Versata provided configuration software to Ford. These solutions included software to support Ford’s online buying site. (Ex. 2006, “Information Week.”) These solutions also included ACM. Ford used ACM to configure vehicles and to provide product configuration data to engineering, manufacturing, sales, marketing, and ordering software that Ford uses around the world. Before ACM, configuration errors caused Ford substantial waste. ACM solved these problems.

The technology in the ’651 patent is one of the foundational technologies in ACM. In particular, the graphically-maintained rules-based approach of the ’651 patent addressed the issues that Ford’s internal configuration software had with handling the complexity and volume of data required to support Ford’s automotive manufacturing needs. (*Id.*, p. 5.)

**B. Versata developed the technology of the '651 patent to address the challenges of maintaining and configuring complex constraint-based systems.**

The '651 patent, entitled “Method and Apparatus for Maintaining and Configuring Systems,” provides a framework for defining systems by defining the components of the system and relationships between the components of the system. ('651 patent, Abstract.)

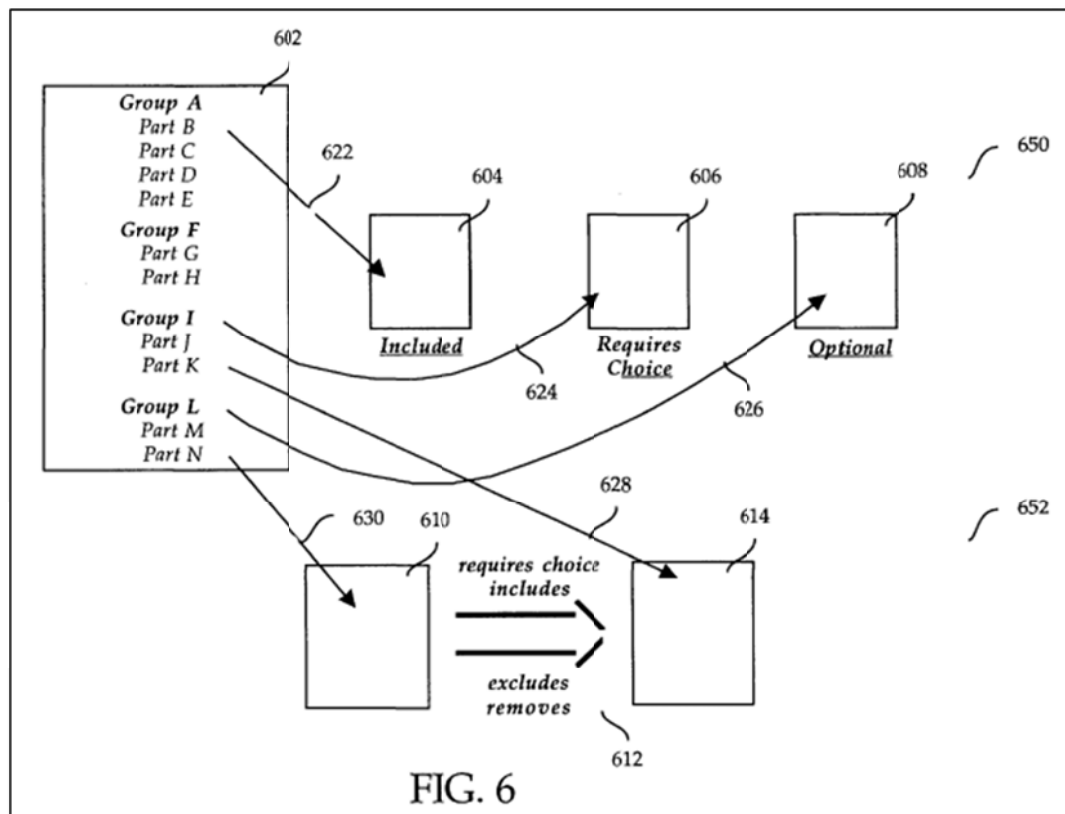


FIG. 6 of the '651 patent, reproduced above, illustrates an example graphical user interface (GUI) implementing the easy-to-use product and part relationships approach. As the '651 patent describes, *products* are initially defined by dragging

parts from pane 602 to one of panes 604, 606, or 608. ('651 patent, 8:8-10.)

Specifically, the '651 patent details:

A user can drag elements from pane 602 to panes 604-608 to define a product. For example, to include Part B in the product definition, Part B is dragged from pane 602 to pane 604. Alternatively, to drag parts B, C, D, and E, group A can be dragged from pane 602 to pane 604. Group A and its component parts (parts B, C, D, and E) are thereby included in the product definition. Similarly, a user can specify that a configuration user must choose a part from a group, e.g., Group I, by dragging one or more parts or a group into pane 606. An optional part or group can be identified by dragging an element, e.g., Group L, into pane 608.

('651 patent, 8:12-22.)

FIG. 6 also illustrates example GUI components used to easily define *part* relationships used in building a particular valid configuration that constitutes a product. ('651 patent, 8:10-11.) As the '651 patent details:

In addition to defining the elements (and their types) that comprise the product, the maintainer can specify relationships between parts of the product. Panes 610 and 614 and relationship 612 are used to define relationships between parts. The maintainer can drag an element (or elements) from pane 602 into pane 610. For example, the user can drag Part N from pane 602 into pane 610. The element(s) dragged into pane 610 are referred to as the left-hand side of the relationship. The element (or elements) from pane 602 that is related to the left-hand

element(s) is dragged into pane 614. The element(s) dragged into pane 614 are referred to as the right-hand side of the relationship. For example, the user can drag Part K from pane 602 to pane 614. ('651 patent, 8:37-49.)

Product relationships are built, by example, by classifying parts in a product definition as “included (parts that are included by default), required choices (a choice among a group of parts that must be made to achieve a valid configuration), optional (parts that can be optionally included in the configuration).” ('651 patent, 2:7-11.) Part relationships, by way of example, can be of the type “included” (the part is included automatically), “excluded” (the part is excluded automatically if its inclusion is invalid), “removed” (when a first part exists, the part is removed), and “requires choice” (a choice is needed from a group of parts). (*Id.*, 2:22-34.)

The ability to build and maintain these relationships graphically is a function of what the '651 patent calls a “maintenance system.” The GUI shown in FIG. 6 of the '651 patent facilitates the actual building and maintenance of the product and part relationships.

**C. Before the '651 patent, maintaining a configuration system required sophisticated programming knowledge to maintain complex, high-latency systems.**

Building and maintaining a configuration system requires, at a basic level, the knowledge of which configuration options are viable in a certain configuration.

The '651 patent explains how, for example, an automobile salesperson might know the various configuration options for a vehicle. This knowledge of acceptable vehicle configurations can be used to build a configuration system that allows an automobile salesperson or prospective buyer to build a valid vehicle product that conforms to configuration requirements. The '651 patent explains:

A system is comprised of components. Before a system can be built the components of the system must be identified. To configure a system, a user must select the parts to include in the system. Typically, one who is knowledgeable about a system and its components defines the system. Thus, for example, an automobile salesperson assists an automobile buyer in determining the type and features of the automobile. The salesperson understands the features and options that are available to create a valid configuration. Some features and options cannot be combined. The selection of some features caused other features to be unavailable, etc. It would otherwise be difficult for the buyer to identify all of the features and options available on the automobile that can be combined to create a valid configuration.

('651 patent, 1:12-25.)

However, at the time of the '651 patent, building and maintaining a configuration system required more than knowledge of configuration options and basic computer usage. Existing approaches relied on complex configuration

languages, and had to be developed and maintained by computer programmers.

The '651 patent explains the challenge thusly:

Computer systems have been developed to assist one in configuring a system. However, these systems use a configuration language to define a system. Like a programming language, a configuration language uses a syntax that must be understood by a user who is maintaining the data (i.e., a data maintainer). To use one of these configuration systems, it is necessary for a data maintainer to understand the configuration language. This limits the number of users who are able to use the configuration systems. That is, the level of sophistication needed to communicate with the configuration system (through a configuration language) results in less sophisticated users being unable to use the system.

('651 patent, 1:26-37.)

**D. To solve the challenge of maintaining and operating complex constraint-based systems, the '651 patent uses easy to maintain “product relationship(s)” and “part relationship(s).”**

The '651 patent sets out to solve the challenge of complex configuration syntax with rules that are simple to both build and understand graphically. These rules operate as product relationships (an association between a product and one or more parts) and part relationships (an association that exists between a first set of parts and a second set of parts).

These associations, which are more simply maintained graphically by a non-programmer, are recited in the claims. For example, claim 60 recites:

60. A method of configuring a system comprising the steps of:  
*generating a definition for said system, said definition containing one or more elements and being conveyed graphically using a set of product relationships;*  
*generating a set of part relationships between said one or more elements, said set of part relationships being conveyed graphically using a set of part relationships; and*  
receiving input from a configuration user;  
validating said input based on said definition, said set of product relationships, said set of part relationships, and a current configuration state; and  
identifying a set of valid configuration options using said definition, said set of product relationships, said set of part relationships and said current configuration state.  
(Emphasis added.)

Ford notes that “prior art literature describing configuration software that was not considered during examination of the ’651 Patent had already recognized, and solved, the alleged problems in the manner described and claimed in the ’651 Patent.” (Pet., p. 5.) But while the ’651 patent acknowledges that “[c]omputer systems have been developed to assist one in configuring a system” (’651 patent,



1:26-27), the '651 patent completely rewrote the script on how to develop such systems.

As described above, the '651 patent addresses the challenge of maintainability of its configuration technology by implementing a simple set of relationship rules that can be managed graphically. Product relationships are built, by example, by classifying parts in a product definition as “included (parts that are included by default), required choices (a choice among a group of parts that must be made to achieve a valid configuration), optional (parts that can be optionally included in the configuration).” ('651 patent, 2:7-11.) Part relationships, by way of example, can be of the type “included” (the part is included automatically), “excluded” (the part is excluded automatically if its inclusion is invalid), “removed” (when a first part exists, the part is removed), and “requires choice” (a choice is needed from a group of parts). ('651 patent, 2:22-34.)

**E. Ford’s cited Axling reference describes a configuration system with the same maintenance complexities that the '651 patent was designed to overcome.**

Ford cites to a paper by Tomas Axling and Seif Haridi (“Axling”) entitled “A Tool for Developing Interactive Configuration Applications.” Axling promises “a less complex and more interactive approach to configuration that we label Interactive Configuration by Selection (ICS).” (Ex. 1004, Axling, p. 2.) Ford

highlights that ICS has “a graphical user interface making the domain model easy to understand and work with.” (Pet., p. 9 (citing Ex. 1004, p. 8.)

In order to allow for maintaining the Axling system, the authors developed a prototype tool named OBELICS (OBject-oriented Language for ICS<sup>3</sup>). (Ex. 1004, pp. 1-2 and 9.) OBELICS is a programming language built on the Prolog programming language. (*Id.*, p. 9.) Axling provides several examples demonstrating the programming knowledge needed in order to develop for and maintain configurations. For example, Axling explains that objects may be written using the following programming syntax:

```
object-identifier :: {  
    sentence-1 &  
    sentence-2 &  
    :  
    sentence-n  
    }.
```

(Ex. 1004, p. 10.)

And Axling further explains that constraints that must be satisfied if a component class is to be considered a valid option are expressed through the use of an “OBELICS constraint expression (OCE).” Axling explains:

---

<sup>3</sup> The term “ICS,” in turn, stands for “Interactive Configuration by Selection.” (Ex. 1004, p. 2.)

An OBELICS constraint expression (OCE) is a conjunction or a disjunction of OCEs or a simple constraint expression. A simple constraint expression is an equality/inequality of attributes accessible in the component class, a linear rational constraint, or a call to an access method. A linear rational constraint is an linear equation of attributes accessible in the component class enclosed in curly brackets, e.g. { @price<@k+100}. An access method is one of OBELICS' predefined methods for accessing other instances, for example:

**instance(CC,Expr,I)** which means “There is an instance I of CC that satisfies Expr”, Expr being of the same type as preconditions. This method enables one to access other instances and their attributes using Expr as criteria.

**no\_instance(CC,Expr)** which means “There does not exist an instance of CC that satisfies Expr”, Expr being of the same type as preconditions.

**apply(F,I,Res)** which means “Res is the result of applying F/2 on the part-of hierarchy rooted at instance I.

(Ex. 1004, p. 12.)

Ford misinterprets the Domain Model Editor of Axling in stating that “[r]elationships between the ‘compsys’ product and its available parts, and relationships between the parts themselves, are depicted in the Domain Model Editor.” (Pet., pp. 10-11.) In Axling’s example of FIG. 5.2, there is not a product

relationship between “compsys” and any parts. Nor is there any relationship between one part and another part.

Axling’s goal is to facilitate the development of the older, difficult to maintain constraint-based systems that the ’651 patent improved upon. Axling describes as a design rationale for ICS:

The tool should support the modeling of the domain. Instead of working with unstructured collections of rules, as in regular rule-based languages, the domain should be structured into chunks of knowledge. The chunks of knowledge should be presented to the user with a graphical user interface, making the domain model easy to understand and work with. The chunks of knowledge should be as independent as possible, letting the user concentrate on one chunk at a time without being concerned about interference with the rest of the domain model.

(Ex. 1004, p. 8.)

When considering Axling, it is important to understand which behavior is actually part of the functionality for building and maintaining configuration options. The actual configuration in Axling is maintained entirely in code and defined by constraints. Axling’s goal, while still using traditional constraint-based processing, is to make the constraint editing process more accessible by allowing a programmer to consider individual chunks of knowledge at a time through the domain model editor.

Axling describes that “[t]he domain model editor presents hierarchies in an understandable way and makes it easy to add and modify components.” (Ex. 1004, p. 12.) Axling does not describe that the addition or modification of components is in any way handled graphically, short of showing the end results in the Domain Model Editor. Instead, Axling illustrates the *class* definitions that need to be *coded* using the OBELICS programming language for each component (*see id.*, p. 12) and its associated constraints and subcomponent classes.

For the example of FIG. 5.2, the accompanying component class for “compsys” is shown in Axling at pp. 15-16. This code fragment shows that the “compsys” component class has several subcomponents, consistent with FIG. 5.2:

```
compsys::{  
    ...  
    subcomponents([(connector,*),  
                    (procs,!),  
                    (soft,*),  
                    (printers,*),  
                    (cage,!),  
                    (dataEntryDevice,!),  
                    (display,!),  
                    (board,*),  
                    (memory,*),  
                    (motherBoard,!),  
                    (externalStorage,+),
```

```
(communicationComponent,*)])  
}.
```

The subcomponents in this code fragment are consistent with the subcomponents shown in FIG. 5.2. Each of these objects, and the syntax used to build these objects, is provided in the format of SICStus Prolog Objects System objects (or “Prolog Objects”).

As Axling notes, “[t]he OBELICS domain model is a form of component hierarchy that describes is-a and part-of relations.” (Ex. 1004, p. 12.) This is evident in the definition of the specific “laserPrinter” component class in Axling, p. 16:

```
laserPrinter::{  
    super(printer)&  
    description('Laser printer')&  
    precondition((@graphic=true,{ @price=8})))&  
    connection(printerToPort,[(laserPrinter,L:self(L)),  
                               (cable,C:true),  
                               (port,P:P@type=serial)])  
}.
```

This definition of the “laserPrinter” component class “imposes constraints on the attributes graphic and price,” which are attributes of the “printer” class which “laserPrinter” is a subclass of. (*Id.*, p. 17.)

It becomes clear that the “compsys” component class, above, which includes “printers” as a subcomponent class, does not define a relationship between a product and parts, or among parts. Instead, the “printers” *class* is what Axling terms a “composite component class,” and the “laserPrinter” *class* is what Axling terms a “primitive component class.” (*Id*, p. 12.)

In order to establish relationships between the “laserPrinter” class and another class, a connection is needed. Axling provides constraints within the description of these connections. And Axling’s constraints are specifically handled at a *class* level, with no understanding of product-to-part or part-to-part relationships. For example, Axling describes the connectivity of a connection class as follows:

**Between:** A set of triples (CC,SC,OSC) that describe the components the connection class can connect. The triples are defined as follows:

CC is a component class.

SC is a set of connection classes.

OSC is a set of connection classes.

Each triple states that a connection of the class must include a component C of class CC that is not included in any other connections except for connections of classes that occur in SC or in OSC if granted by the user.

(Ex. 1004, pp. 14-15.)

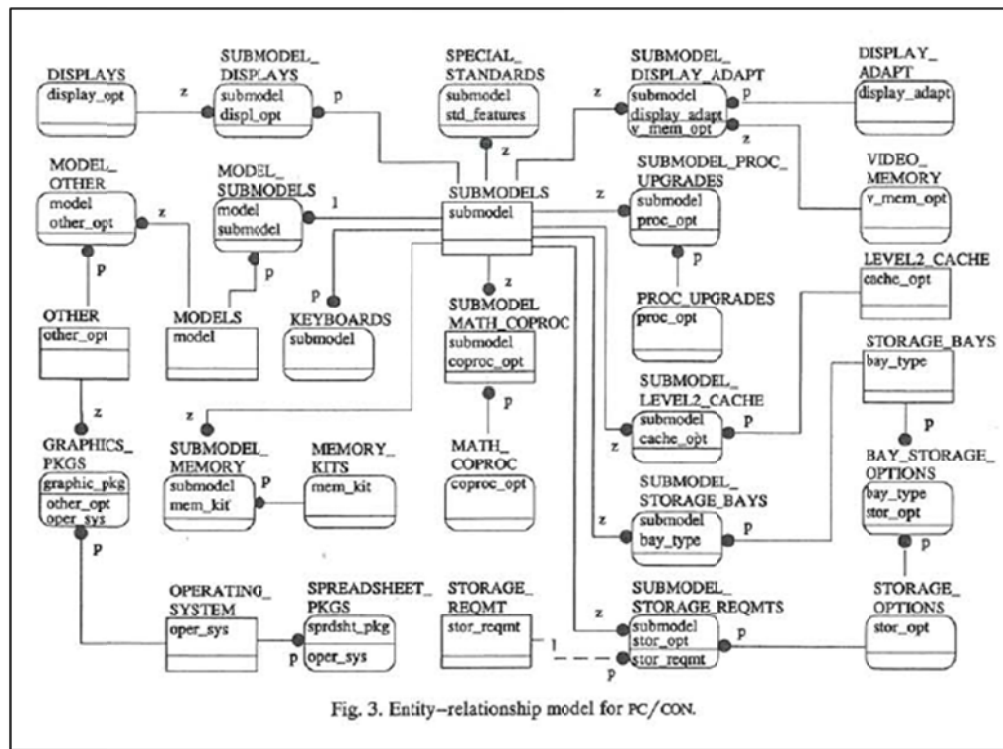
The connection classes of Axling therefore require complex programming ability in order to define the constraints of each of the connection classes. This is evident from Axling dealing in describing possible connections between components not using rules-based relationships that can be maintained graphically, but rather indirectly through constraints on the component classes that must be maintained through code, noting that “[i]n a connection class we specify which component classes are connected, and for each component class in the connection class, we specify whether a component instance may occur in other connection instances.” (Ex. 1004, p. 14.)

Axling is nothing like Ford’s picture of an easy-to-maintain graphical configuration system. Rather, Axling is simply another iteration of the complex and challenging to maintain systems that the ’651 patent improved upon.

**F. Ford’s cited Fohn reference likewise suffers from the same maintenance complexities that the ’651 patent was designed to overcome.**

Fohn is concerned with the need for “maintainability and expandability of existing configuration information models.” (Ex. 1005, Fohn, p. 7.) This is a challenge for systems like Fohn because, similar to Axling, the architecture of Fohn’s Saturn is described as “a constraint-system shell in which problems can be modeled as constraints and exercised through interactive constraint satisfaction [13].” (Ex. 1005, p. 8.)





The architecture of Fohn maintains product data taken from the IBM Sales Advisor catalogue and reorganized into the relational model shown in FIG. 3, reproduced above. The data formatted in accordance with this relational model is then stored in a relational database, which is accessed and manipulated in accordance with a set of constraints drawn from a constraint-based model shown in FIG. 2 of Fohn.

The information stored in the database of Fohn only makes sense in the context of its constraints. As Fohn notes:

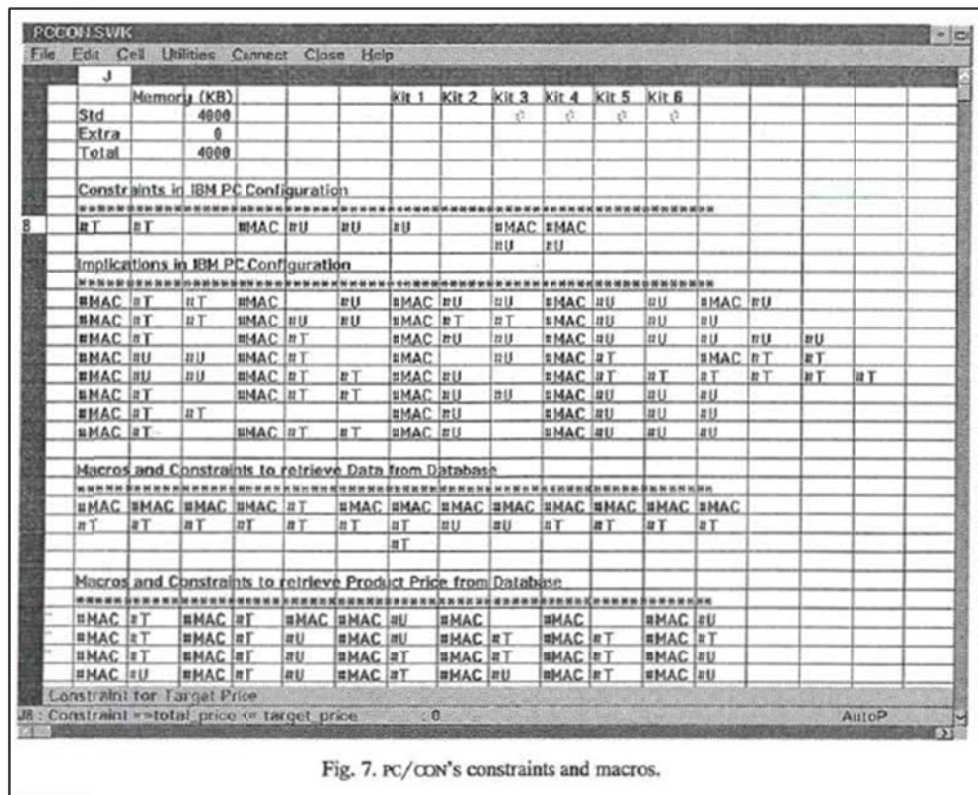
Part of the constraint sheet shown in Fig. 4 holds the final computer configuration as groups of attributes that display a hardware configuration, software configuration, and price that satisfy both

customer and manufacturing requirements. Another region of the constraint sheet holds constraints that dynamically and interactively manage the process of finding a valid computer configuration (refer to [5] for specific details of the syntax and operation of Saturn's constraints).

(Ex. 1005, p. 11.)

In order to edit these constraints, and therefore the rules by which a valid computer configuration is built, Fohn provides a section of the display called the "Macro and Constraint region" of the constraint sheet. This section of the constraint sheet, shown in FIG. 7 of Fohn and reproduced below, allows a programmer or other developer to utilize this facility to examine or edit the macro or constraint.

In FIG. 7, the "#MAC" on screen "represents a macro which can be used to write Prolog-like subprograms to perform database retrieval/updates and other operations." (Ex. 1005, p. 15.)



Contrary to the manner in which Ford has presented Fohn, it does not provide any significant ease of use in the actual maintenance of configuration information. Instead, just like Axling, a software developer, likely versed in Prolog, is needed in order to perform any significant maintenance of the system.

**G. The combination of Axling and Fohn represents nothing more than the complex, programmer-maintained constraint-based systems the '651 patent innovated upon.**

The difference between the simplified rules-based approach of the '651 patent and the complex systems of Axling and Fohn is evident in how a configuration is built. In a constraint-based system such as Axling or Fohn, the constraints provide a programmatic basis for 'how' or 'why' components go

together. In contrast, the rule-based relationships of the '651 patent address 'what' components go together, and is readily conveyed and maintained graphically.

Axling relies on constraint-based configuration, and provides tools for managing these constraints. Axling's hierarchical approach facilitates the review of individual constraints at different parts of the hierarchy, but nevertheless operates on complicated constraint-based programming that requires specific understanding of SICStus Prolog Objects in order to write. While the Domain Model Editor purports to enable GUI-based editing, this hierarchical display only facilitates jumping to a particular section of object code for ease of programming.

Fohn is similarly representative of the commonplace constraint-based systems that were in regular use prior to the '651 patent's rule-based processing. Fohn's Saturn system requires the specific use of "constraints that dynamically and interactively manage the process of finding a valid computer configuration." These constraints must be modified in order to change the operation of the configuration system.

### **III. CLAIM CONSTRUCTION**

Versata contends that there are several claim terms requiring construction. Construction of the terms "product relationship(s)" and "part relationship(s)" under the *Phillips* standard is required in this proceeding.

Claim Term	Ford's Construction	Versata's Construction
“product relationship(s)”	plain and ordinary meaning under broadest reasonable interpretation	<i>An association between a product and one or more parts, the association having a left-hand side and a right-hand side. The product represents the left-hand side of the relationship, and the set of elements represents the right-hand side of the relationship.</i>
“part relationship(s)”	plain and ordinary meaning under broadest reasonable interpretation	<i>An association that exists between a first set of parts and a second set of parts, the association having a left-hand side and a right-hand side. The first set of parts represents the left-hand side of the relationship and the second set of parts represents the right-hand side of the relationship.</i>

Ford's assertion that “Petitioner does not believe any terms in the challenged claims require construction beyond their plain and ordinary meaning under the broadest reasonable interpretation standard required for this proceeding” (Pet., p. 16) is seemingly impossible to reconcile with the remainder of the petition, as Ford draws heavily from the 2005 *Selectica* litigation in discussing the terms “product relationship” (Pet., p. 26) and “part relationship” (Pet., p. 27).

#### **A. Relevant law**

In a Decision entered July 14, 2016 (Paper No. 9), the Board granted Versata's “Motion for District Court (*Phillips*) Claim Construction” (Paper No. 8). Accordingly, the claim construction standard for this *inter partes* review

proceeding is equivalent to that set out in prior district court litigation between the parties.

While much ink has been spilled in recent months regarding the difference, if any, in the application of the broadest reasonable interpretation (“BRI”) standard and the district court *Phillips* standard<sup>4</sup>, any differences need not be resolved in order to address claim construction at this stage. The claim terms presented below have been resolved through agreement of the parties in the district court under *Phillips*.

---

<sup>4</sup> For example, in an amicus brief in *Cuozzo*, Judge Michel (ret.) opines (pp. 5-6), “The approach of construing claims so that they receive the meaning a skilled artisan would give them in the context of the entire specification is also supposed to be used by courts for issued patents. *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1316-17 (Fed. Cir. 2005) (*en banc*). Thus, the putative claim construction standard between courts and the Patent Office is the same—with the one minor difference being that courts may apply disclaimers made by a patent owner during prosecution, even if the claims do not supply a textual hook for such disclaimer, while the Patent Office properly forces an applicant or Patent Owner to put the explicit text in the claims if it is not already there.”

In particular, the parties have agreed that the Eastern District of Texas construction of the terms “product relationship(s)” and “part relationship(s)” from the *Selectica* litigation (*Trilogy v. Selectica*, 405 F. Supp.2d 731 (E.D. Tex. Dec. 20, 2005)) should apply in the current dispute in the Eastern District of Michigan. Since the parties have agreed to the construction of these terms in the district court, and have done so under the same claim construction standard applied in this *inter partes* review proceeding, the following constructions should control here.

**B. “product relationship(s)”**

Claim 60 recites, in part, “generating a definition for said system, said definition containing one or more elements and being conveyed graphically using a set of *product relationships*.” (’651, 28:65-67 (emphasis added).) The district court in *Selectica* construed the term “product relationship(s)” to mean:

[A]n association between a product and one or more parts, the association having a left-hand side and a right-hand side. The product represents the left-hand side of the relationship, and the set of elements represents the right-hand side of the relationship.

(Ex. 1019, CC Order, p. 11.)

This construction is appropriate in this *inter partes* review proceeding, as it is the construction agreed to by both parties in the pending Eastern District of Michigan litigation.

Ford notes that Versata advanced the construction “[a] classification of the specific association that exists between a product and one or more parts” in the *Selectica* litigation. (Pet., p. 26.) Even if this construction is used instead, the association between a product and one or more parts is present as in the construction adopted by the district court, which is the subject of the discussion below.

**C. “part relationship(s)”**

Claim 60 recites “generating a set of *part relationships* between said one or more elements.” (’651 patent, 29:1-2 (emphasis added).) The district court in *Selectica* construed the term “part relationship(s)” to mean:

[A]n association that exists between a first set of parts and a second set of parts, the association having a left-hand side and a right-hand side. The first set of parts represents the left-hand side of the relationship and the second set of parts represents the right-hand side of the relationship.

(Ex. 1019, CC Order, p.11.)

This construction is appropriate in this *inter partes* review proceeding, as it is the construction agreed to by both parties in the pending Eastern District of Michigan litigation.

Ford notes that Versata advanced the construction “[a] classification of the specific association that exists between a first set of one or more parts and a second



set of one or more parts” in the *Selectica* litigation. (Pet., p. 27.) Even if this construction is used instead, the association between a first set of parts and a second set of parts is present as in the construction adopted by the district court, which is the subject of the discussion below.

#### **IV. ARGUMENT**

“In an *inter partes* review, the burden of persuasion is on the petitioner to prove ‘unpatentability by a preponderance of the evidence’ . . . and that burden never shifts to the patentee.” *In re Magnum Oil Tools Int’l, Ltd.*, No. 2015-1300, 2016 WL 3974202, p. 6 (Fed. Cir. July 25, 2016) (citing *Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015)). Here, for the reasons that follow, Ford has not met its burden of persuasion. The Board should therefore deny institution.

##### **A. Ford failed to show that the combination of Axling and Fohn teaches or suggests the claimed “product relationships.”**

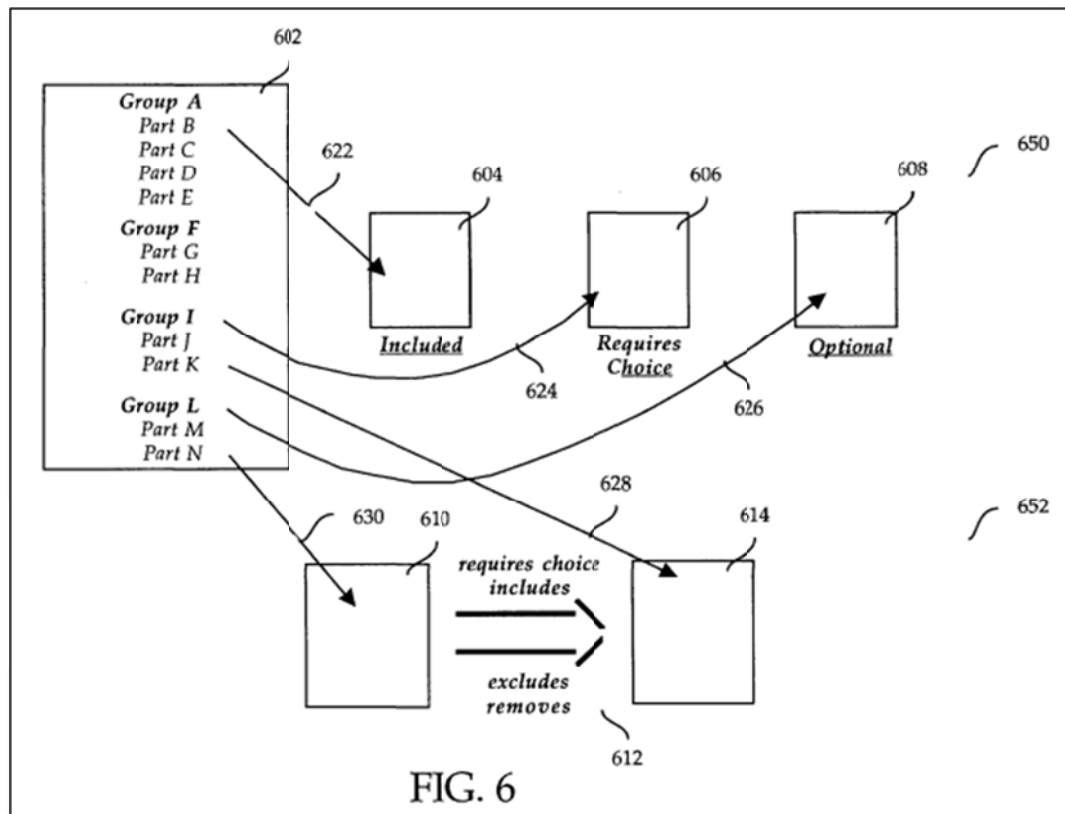
Claim 60 of the ’651 patent recites “generating a definition for said system, said definition containing one or more elements and being conveyed graphically using a set of product relationships.” Neither Axling nor Fohn, alone or in combination, teaches or suggests the claimed “product relationships.”

As described in the claim construction section, the claimed “product relationships” require an association between a product and one or more parts. Axling and Fohn lack any such association.

Ford relies exclusively on Axling for this teaching. To disguise Axling’s deficiency in teaching or suggesting an association between a product and one or more parts, Ford does two things. *First*, Ford blurs the line between a **part** in the ’651 patent and a **class** in Axling. *Second*, Ford completely glosses over the constraint-based relationships in Axling that are handled with regard to classes rather than parts.

Ford argues that Axling’s “Domain Model Editor in Figure 5.2 discloses an association between the product ‘compsys’ on the left, and its constituent parts on the right: ‘cage,’ ‘procs,’ ‘soft,’ ‘printers,’ etc.” and that “[t]he relationship itself is depicted, using the symbols !, \*, ? and + described above.” (Pet., p. 27.) But rather than being **parts** as Ford alleges, “cage,” “procs,” “soft,” “printers,” etc. are **classes** in Axling.

To recap, FIG. 6 of the ’651 patent illustrates how a user of a back-end maintenance and configuration system GUI can establish part relationships between a product and its parts, by way of example. Specifically, the upper section of the GUI marked 650 is the “product definition section” that allows the user to build desired product relationships.



As the '651 patent describes, *products* are defined by dragging parts from pane 602 to one of panes 604, 606, or 608. ('651 patent, 8:8-10.) Specifically, the '651 patent details:

A user can drag elements from pane 602 to panes 604-608 to define a product. For example, to include Part B in the product definition, Part B is dragged from pane 602 to pane 604. Alternatively, to drag parts B, C, D, and E, group A can be dragged from pane 602 to pane 604. Group A and its component parts (parts B, C, D, and E) are thereby included in the product definition. Similarly, a user can specify that a configuration user must choose a part from a group, e.g., Group I, by dragging one or more parts or a group into pane 606. An optional part

or group can be identified by dragging an element, e.g., Group L, into pane 608.

(’651 patent, 8:12-22.)

In contrast, Axling lacks any relationships between parts and products. Ford analogizes the “compsys” class of Axling to the product, and uses the hierarchy shown in Axling as the basis for its comparison to the ’651 patent.

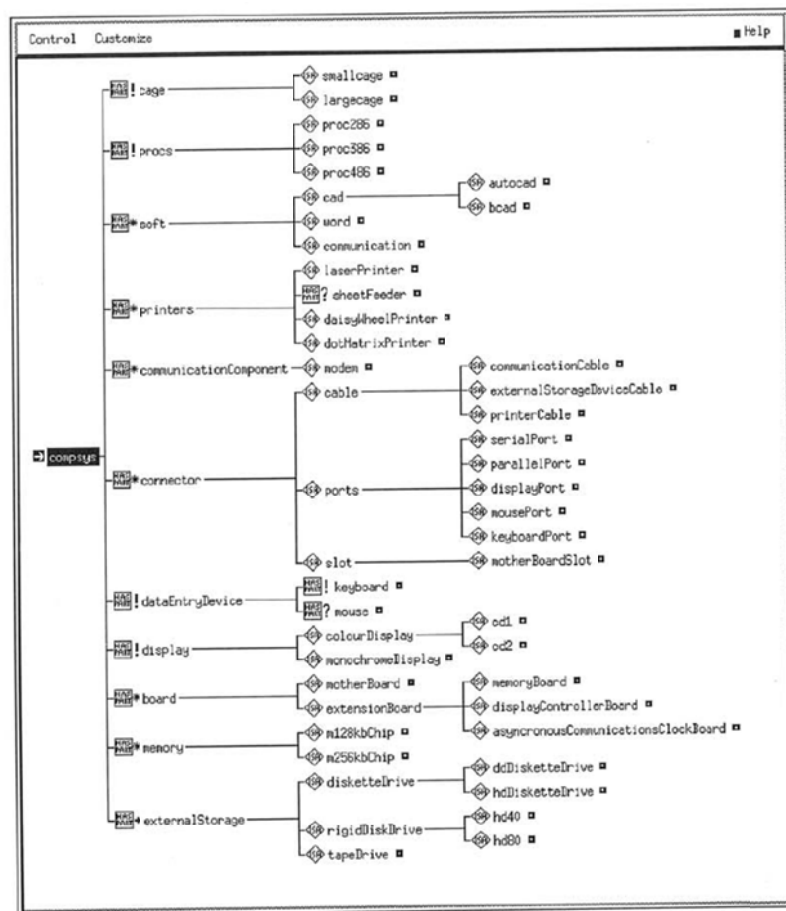


FIGURE 5.2. The Domain Model Editor

For the example of FIG. 5.2, reproduced above, the accompanying component class for “compsys” is shown in Axling (Ex. 1004, pp. 15-16.) This

code fragment shows that the “compsys” component class has several subcomponents, consistent with FIG. 5.2:

```
compsys::{  
    ...  
    subcomponents([(connector,*),  
                    (procs,!),  
                    (soft,*),  
                    (printers,*),  
                    (cage,!),  
                    (dataEntryDevice,!),  
                    (display,!),  
                    (board,*),  
                    (memory,*),  
                    (motherBoard,!),  
                    (externalStorage,+),  
                    (communicationComponent,*)])  
    }.
```

The subcomponents in this code fragment are consistent with the subcomponents shown in FIG. 5.2. Each of these objects, and the syntax used to build these objects, is provided in the format of SICStus Prolog Objects System objects (or “Prolog Objects”).

As Axling notes, “[t]he OBELICS domain model is a form of component hierarchy that describes is-a and part-of relations.” (Ex. 1004, p. 12.) This is

evident in the definition of the specific “laserPrinter” component *class* in Axling(Ex. 1004, p. 16):

```
laserPrinter::{  
    super(printer)&  
    description('Laser printer')&  
    precondition((@graphic=true,{ @price=8})))&  
    connection(printerToPort,[(laserPrinter,L:self(L)),  
                                (cable,C:true),  
                                (port,P:P@type=serial)])  
    }.
```

This definition of the “laserPrinter” component “imposes constraints on the attributes graphic and price,” which are attributes of the “printer” *class* which “laserPrinter” is a *subclass* of. (*Id*, p. 17.)

It becomes clear that the “compsys” component *class*, above, which includes “printers” as a subcomponent *class*, does not define a relationship between “compsys” and any *parts*. Instead, the “printers” class is what Axling terms a “composite component class,” and the “laserPrinter” class is what Axling terms a “primitive component class.” (*Id*, p. 12.)

In order to decipher the options available for subcomponents of the “printers” class, a set of symbols is applied as a constraint to each of the subcomponents. For example, the symbol “\*” shown next to the “printers” class

applies a constraint to indicate that zero or more subcomponents from the “printers” class may be included. This is handled without establishing any relationships to any *specific* printer components (such as a laser printer).

As a result, Ford’s argument that “compsys” is analogous to a ’651 patent *product*, and “printers” (or “cage,” “procs,” or “soft”) is analogous to a ’651 patent *part*, holds no water. The “printers” *class* of Axling is simply a subcomponent *class* of “compsys” in which several constraints relating to printers *in general* can be programmed.

To wit, *classes* in Axling are abstract families, whereas *parts* would be very specific items that belong to a class. For example, a class could be “printers” with a number of specific printers (parts) defined by that class -- e.g., “HP LaserJet 9000,” “Canon PIXMA Pro-100,” “Brother HL-2200.” In focusing on class-based constraints, Axling provides absolutely no disclosure regarding a relationship between the purported product, “compsys,” and any *part* whatsoever.

Moreover, what Ford compares to the actual “product relationship” itself, the symbols !, \*, ? and + applied to the subcomponent classes of “compsys,” is a constraint applied to subcomponent classes of the subcomponent classes. To wit, the “!” constraint applied to the “printers” subcomponent class applies that constraint to any of its subcomponent classes.

The lack of any relationship between the purported product, “compsys” and any parts is evident in the effect the “\*” constraint has on the “printers” subcomponent. Specifically, if a new printer subcomponent class is added beneath the “printers” subcomponent class, the “\*” constraint is completely agnostic to the existence of any new part.

In contrast, the ’651 patent’s product relationships do not operate on constraints, but rather rules. Product relationships in the ’651 patent are built, by example, by classifying parts in a product definition as “included (parts that are included by default), required choices (a choice among a group of parts that must be made to achieve a valid configuration), optional (parts that can be optionally included in the configuration).” (’651 patent, 2:7-11.) Every single part included in a product is itemized as a product-to-part relationship in accordance with this approach. This is simply not the case for Axling.

As noted above, Ford does not rely on Fohn to teach or suggest this claim feature. Thus, in view of the foregoing, Ford has failed to support its allegation that the combination of Axling and Fohn teaches or suggests “generating a definition for said system, said definition containing one or more elements and being conveyed graphically using a set of product relationships” as recited in claim 60.



Claims 61-72 depend from claim 60. Each of Ford's grounds relating to these dependent claims relies on this application of Axling and Fohn to claim 60. Therefore, all of Ford's grounds should be denied.

**B. Ford failed to show that the combination of Axling and Fohn teaches or suggests the claimed “generating a set of part relationships between said one or more elements.”**

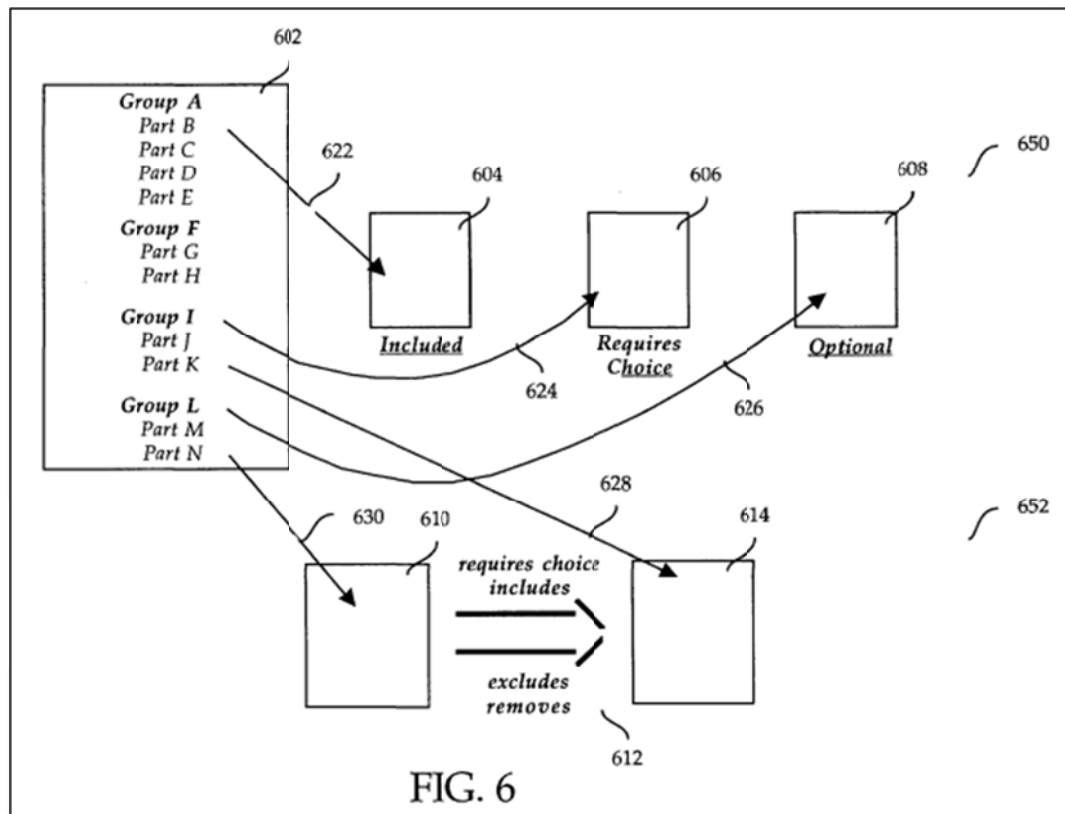
Claim 60 of the '651 patent recites “generating a set of part relationships between said one or more elements, said set of part relationships being conveyed graphically using a set of part relationships.” Neither Axling nor Fohn, alone or in combination, teaches or suggests the claimed “part relationships.”

As discussed above in the claim construction section, “part relationships” require an association between a first set of parts and a second set of parts. Axling and Fohn lack any such association.

Ford relies exclusively on Axling for this teaching. Similar to the discussion above regarding “product relationships,” to disguise Axling's deficiency in teaching or suggesting “part relationships” -- an association between a product and one or more parts -- Ford does two things. *First*, Ford blurs the line between a *part* in the '651 patent and a *class* in Axling. *Second*, Ford completely glosses over the constraint-based relationships in Axling that are handled with regard to classes rather than parts.

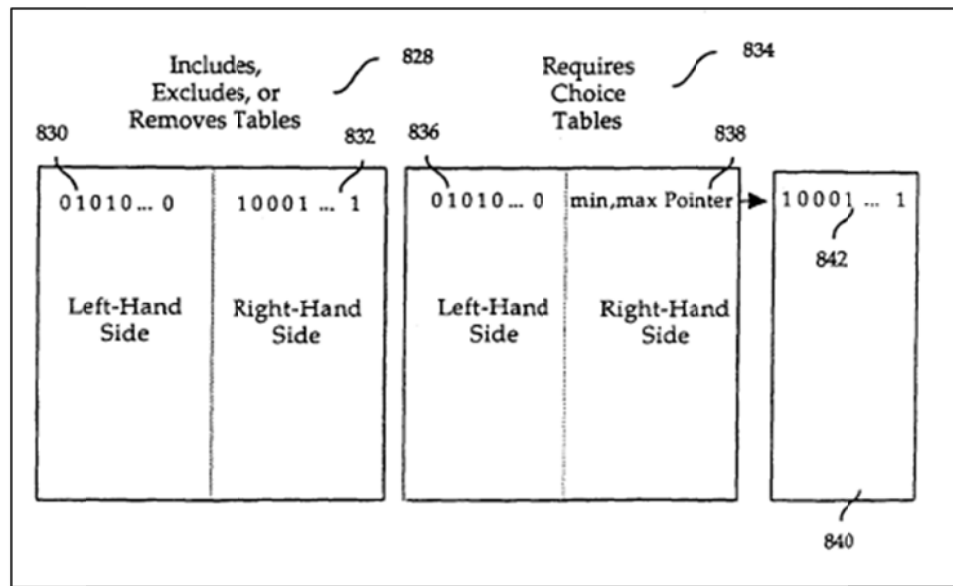
Ford argues that Axling discloses generating “part relationships” by noting that, “[s]imilar to the relationships between the ‘compsys’ system and its component parts described above, the Domain Model Editor permits a user to define and graphically display relationships between the parts themselves.” (Pet., p. 27.) Ford adds that “[f]or example, the part ‘printer’ may optionally include the part ‘sheetFeeder.’” (Pet., p. 27.) But rather than being a *part* as Ford alleges, “printers” is a *class* in Axling. Moreover, any relationship between the “printers” class and the “sheetFeeder” subcomponent class is defined by way of a “connection class,” which connects *component classes*, not *parts* (and does so using a complex constraint-based relationship). (Ex. 1004, p. 14.)

To recap, FIG. 6 of the ’651 patent illustrates how a user of a back-end maintenance and configuration system GUI can establish part relationships between one or more elements, by way of example. Specifically, the lower section of the GUI marked 652 is the “part relationship definition section” that allows the user to build desired part relationships.



As the '651 patent notes, “[p]anes 610 and 614 and relationship 612 are used to define relationships between parts.” ('651 patent, 8:39-41.) Elements (parts or groups) dragged from pane 602 into pane 610 define the left-half of a relationship, and elements dragged from pane 602 into pane 614 define a right-half of the relationship. The '651 patent then notes that “[t]o identify the type of relationship that exists between the left-hand side and right-hand side elements, the maintainer chooses one of the four *part-to-part relationships* from relationship 612.” ('651 patent, 8:50-53 (emphasis added).)

Figure 8B of the '651 patent, reproduced in part below, shows one such exemplary data structure:



In fact, any reference to part-to-part relationships in the '651 patent follows the particular structure of creating a specific association between the parts themselves.

Ford relies on the Domain Model Editor of Axling to teach or suggest the aforementioned claim feature. However, Axling operates in a fundamentally different matter, and does not create any part-to-part relationships. Axling describes that "[t]he domain model editor presents hierarchies in an understandable way and makes it easy to add and modify components." (Ex. 1004, p. 12.) Axling does not describe that the addition or modification of components is in any way handled graphically, short of showing the end results in the Domain Model Editor. Instead, Axling illustrates the *class* definitions that need to be *coded* for each component (*see id*) and its associated constraints and subcomponent classes.

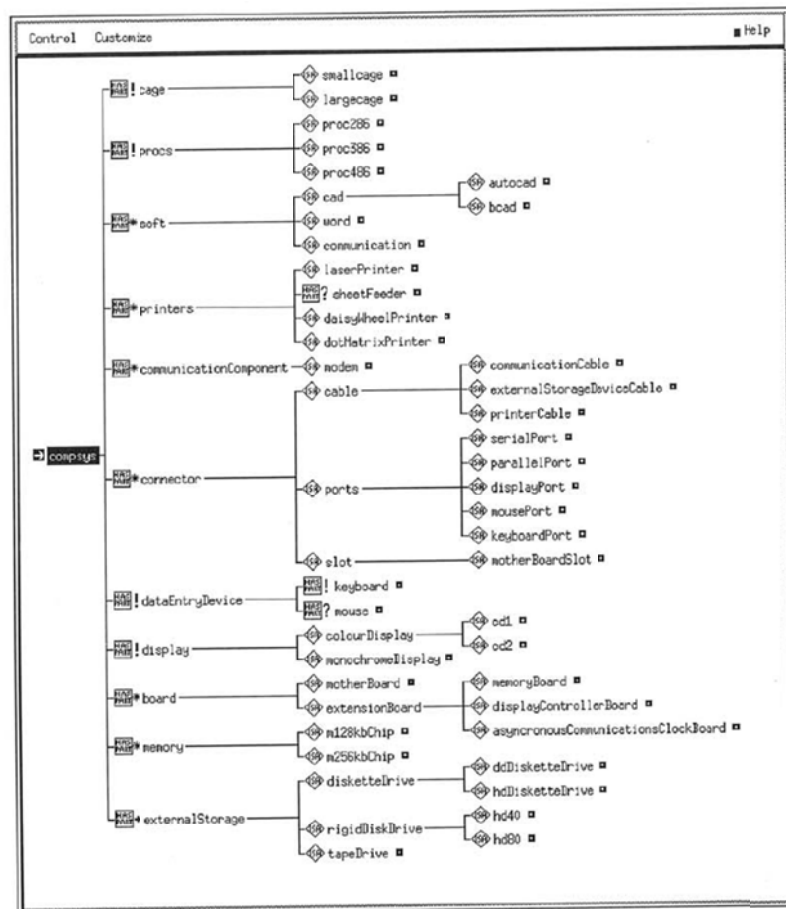


FIGURE 5.2. The Domain Model Editor

Figure 5.2 of Axling, reproduced above, shows an exemplary view of the Domain Model Editor. Ford argues that this particular definition shows elements such as “cage,” “procs,” “soft,” and “printers.” (Pet., pp. 25-26.) Various symbols ?, \*, !, and + are used to convey relationships.

Specifically, Ford notes that “the Domain Model Editor in Figure 5.2 discloses associations between a first set of parts on the left (e.g., “printers,” “dataentrydevice,” etc.) and a second set of parts on the right (“sheetFeeder,”

“keyboard,” “mouse,” etc.).” (Pet., p. 28.) But “printers” and “dataentrydevice” are *classes* used to apply constraints through the use of a *connection class*, and are not themselves *parts*. (Ex. 1004, pp. 15-16.) As Axling notes, “[t]o describe possible connections between components we use connection classes,” adding that “[i]n a connection class we specify which component *classes* are connected.” (Ex. 1004, p. 14.)

As noted above, Ford does not rely on Fohn to teach or suggest this claim feature. Thus, in view of the foregoing, Ford has failed to support its allegation that the combination of Axling and Fohn teaches or suggests “generating a set of part relationships between said one or more elements, said set of part relationships being conveyed graphically using a set of part relationships” as recited in claim 60.

Claims 61-72 depend from claim 60. Each of Ford’s grounds relating to these dependent claims relies on this application of Axling and Fohn to claim 60. Therefore, all of Ford’s grounds should be denied.

**C. The combination of Axling and Fohn fails to teach or suggest “said input is made without regard for the order” as recited in claim 71.**

Dependent claim 71 recites, in part, “said input [from a configuration user] is made without regard for the order in which said one or more elements are selected for said definition.” Ford argues that Axling teaches this feature by allowing “the user to alter previously made configuration decisions.” (Pet., p. 51.)

However, this behavior of Axling does not teach or suggest that an “input is made without regard for the order,” as claim 71 requires, but merely suggests that the user can back up to an earlier point in the ordered process to make changes.

To support its obviousness contention, Ford must explain why a person of ordinary skill in the art would modify and apply Axling in this manner. *See, e.g., KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 418 (2007) (“[I]t can be important to identify a reason that would have prompted a person of ordinary skill in the relevant field to combine the elements in the way the claimed new invention does.”); *see also Travelocity.com L.P. v. Cronos Techs. LLC*, CBM2014-00082, Paper No. 10, p. 20 (P.T.A.B., Sept. 15, 2014). Ford has not done this. Indeed, Ford is entirely silent about why or how the ability to “alter previous choices” teaches or suggests the claimed input being “made without regard for the order in which said one or more elements are selected for said definition.

Thus, for the reasons set forth above, Ford’s petition with respect to claim 71 should be denied.

**D. Ford failed to show that Axling and Skovgaard are printed publications that qualify as prior art.**

Ford asserted, without support or justification, that the Axling and Skovgaard references are printed publications and, therefore, qualify as prior art under 35 U.S.C. § 102. But Ford has not met its burden for establishing that the

Axling and Skovgaard references are prior art. The Board should, therefore, deny Ford's relevant asserted grounds on this basis alone.

It is well-established that the petitioner bears the burden of demonstrating that the references making up its grounds of rejection do, in fact, constitute prior art. In IPRs, "the petitioner shall have the burden of proving a proposition of unpatentability by a preponderance of the evidence." 35 U.S.C. § 316(e). As part of this burden, the petitioner must establish that the applied references qualify as prior art under 35 U.S.C. § 102. *See Drinkware, LLC. v. National Graphics, Inc.*, 800 F.3d 1375 (Fed. Cir. 2015); *Ariosa Diagnostics, Inc. v. Illumina, Inc.*, IPR2014-01093, paper no. 69, pp. 15-16 (P.T.A.B. January 7, 2016). One of the key elements to qualifying as prior art is that the reference must be a printed publication.

It has long been held that a reference is proven to be a "printed publication" only "upon a satisfactory showing that such document has been disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art, exercising reasonable diligence, can locate it." *In re Wyer*, 655 F.2d 221, 226 (C.C.P.A. 1981). Furthermore, the party seeking to introduce a reference "should produce sufficient proof of its dissemination or that it has otherwise been available and accessible to persons concerned with the art to which the document relates and thus most likely to avail themselves of its



contents.” *Id.* at 227. Ford has not done so for either the Axling or Skovgaard references.

As the Board has previously stated, “Federal Rules of Evidence 901 requires that the proponent produce evidence sufficient to support a finding that an item is what the proponent claims it is.” *Standard Innovation Corp. v. Lelo, Inc.*, IPR2014-00148, Paper No. 41, p. 10 (P.T.A.B. April 23, 2015). Here, neither Axling nor Skovgaard exist in the manner Ford has described, and as such their authentication and publication dates cannot be confirmed. Ford has provided no facts to support its allegation that these references are prior art, and without such facts, reliance on these references constitutes a violation of the Federal Rules of Evidence. Thus, the Board should deny institution of this IPR.

### **1. Axling**

Ford cites Axling as having been published in The Journal of Logic Programming, Volume 19. Ford does not provide any support for this beyond stating that “Axling was published in the Journal of Logic Programming publication in 1994 and qualifies as prior art under 35 U.S.C. § 102(a) and (b).” Pet. at 1. Ford’s expert parrots these statements in his declaration. (Ex. 1002, Greenspun declaration, ¶ 51). Simply stated, the Axling article does not exist in the identified publication.

Submitted herewith is a copy of the entire 19th Volume of the Journal of Logic Programming, dated in 1994, as Exhibit 2007. Of the 719 pages in that volume, there is not a single article by Axling contained therein, let alone the specific article submitted by Ford, as the table of contents quickly reveals.

Thus, all that remains available to verify the publication of the Axling reference is the information printed on the article itself. Printed in the header of the Axling reference is “J. LOGIC PROGRAMMING 1994:19, 20:1-679.” This label is problematic for multiple reasons. First, the label identifies the 1994 volume 19/20 of the Journal of Logic Programming. This is the same journal relied upon by Ford, and which Versata has already shown does not contain the relevant article. Second, the label identifies the relevant pages as 1-679. However, the article is only 22 pages long.

Meanwhile, the footer of the submitted Axling article includes printed copyright information: “THE JOURNAL OF LOGIC PROGRAMMING © Elsevier Science Inc., 1994.” However, it has long been held that a printed copyright is insufficient to authenticate a reference. Specifically, because the Federal Rules of Evidence are applicable in Inter Partes Reviews, “copyright ‘dates imprinted on...documents are hearsay when offered to prove the truth of the matter asserted.’” *Standard Innovation Corp.*, IPR2014-00148, Paper No. 41, p. 17. As

such, the copyright date imprinted on Axling cannot be relied upon to establish its purported publication date.

Ford cited to a journal that does not include the subject reference, and did not provide any factual evidence to support the its purported publication date. Additionally, the information on the face of the Axling reference is erroneous and hearsay. For all of these reasons, Ford has failed to satisfy its burden to demonstrate that Axling constitutes prior art to the '651 Patent.

## **2. Skovgaard**

Ford states, without any explanation, that “Skovgaard 1995” qualifies as prior art under 35 U.S.C. § 102(a) and (b). (Pet., p. 2.) But Ford does not even provide a publication date for Skovgaard 1995. Rather, in the bibliographical information, Ford merely lists the date range that is stamped on the face of the reference: January 30 – February 3, 1995. This should have been immediately suspect to Ford, as a publication date cannot be a date range.

Further, a closer look at the Skovgaard 1995 reference makes clear that the date range is not a date of publication. Specifically, the face of the Skovgaard 1995 reference states “Paper presented at Third International Conference ilce’95 Concurrent Engineering & Technical Information Processing January 30 – February 3, 1995.” Thus, the dates listed on the reference appear to be the dates of the conference. And although the paper was purportedly presented at that

conference, there is no indication as to whether the paper was “published” at any time during this or any other period.

The Federal Circuit has previously stated that conference presentation materials are not sufficient as printed publications. *In re Klopfenstein*, 380 F. 3d 1345, 1348-1349 (Fed. Cir. 2004). “The statutory phrase ‘printed publication’ has been interpreted to mean that before the critical date the reference must have been sufficiently accessible to the public interested in the art; dissemination and public accessibility are the keys to the legal determination whether a prior art reference was ‘published.’” *In re Cronyn*, 890 F.2d 1158, 1160 (Fed. Cir. 1989). The determination as to whether a reference qualifies as a printed publication “involves a case-by-case inquiry into the facts and circumstances surrounding the reference’s disclosure to members of the public.” *In re Klopfenstein*, 380 F.3d at 1350.

The Board has previously found papers presented at conferences to be printed publications. For example, in IPR2013-00544, the Board determined that a paper presented to 100-150 people and later published in a book containing all the papers presented at the conference constituted a printed publication. In a similar Federal Circuit case, a paper presented to 50-500 people and disseminated to at least six people was determined to be a printed publication. *MIT v. AB Fortia*, 774 F.2d 1004 (Fed. Cir. 1985).

In the present case, Ford has made no showing as to the number of people the paper was presented to, has made no indication that the paper was disseminated to any number of people, or that it was published with other papers from the conference. Ford hasn't even established a date of publication for the Skovgaard 1995 reference. Without having made any factual showing as to the dissemination and/or public accessibility of the paper, Ford has not established that Skovgaard 1995 constituted a printed publication, nor that it was published prior to the effective priority date of the patent at issue.

**E. Ford's petition lacks a proper obviousness analysis under *KSR* and *Graham*.**

The analysis required to establish obviousness is well-known. *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). To establish obviousness under § 103, “the scope and content of the prior art are to be determined; differences between the prior art and the claims at issue are to be ascertained; and the level of ordinary skill in the art are resolved.” *Graham v. John Deere Co.*, 383 U.S. 1, 17 (1966). This is because a patent composed of several elements is not proven obvious “merely by demonstrating that each of its elements was, independently, known in the prior art.” *KSR*, 550 U.S. at 418 (referring to *United States v. Adams*, 383 U.S. 39 (1966), a companion case to *Graham*). So merely pointing to the presence of all

claim elements in the prior art is not a complete statement of a rejection for obviousness.

In view of the well-settled law, the Board has repeatedly required a showing of the *Graham* factors to institute an *inter partes* proceeding under § 103. For instance, the Board ruled that “[w]ithout having identified specifically the differences between the recited invention and the prior art, Petitioner has failed to make a meaningful obviousness analysis under 35 U.S.C. § 103(a), and, thus, has failed to establish that it is more likely than not that [claims at issue are] unpatentable over the asserted [combination].” *Travelocity.com*, CBM2014-00082, Paper No. 10, p. 20. In particular, the Board requires a petitioner to (1) articulate the differences between the applied references and the claim limitations and (2) “show what reason or reasons a person of ordinary skill in the art would have to combine the limitations allegedly taught by each one of the combined references to achieve the recitations of [the] claims.” *Id.*, p. 19. Ford has made neither of these showings.

**1. Ford failed to articulate the differences between the applied art and the claimed subject matter.**

In its entire petition, Ford never explains the *claim features* that are lacking from one of its two asserted references (*i.e.*, Axling and Fohn) and that are allegedly taught by the other reference, as required by *Graham*. This omission is

not trivial. To the contrary, “the assessment of differences between the prior art and the claimed subject matter . . . prevents evaluation of the invention part by part,” which avoids the importation of “hindsight into the obviousness determination by using the invention as a roadmap to find its prior art components.” *Princeton Biochemicals, Inc. v. Beckman Coulter, Inc.*, 411 F.3d 1332, 1337 (Fed. Cir. 2005).

Ford simply asserts that the combination of references disclose the claims, without any discussion as to what is lacking between the references. Additionally, Ford relies on the knowledge of a person having ordinary skill in the art to fill the gaps in their analysis without clearly demarcating what is lacking from the references that require that additional knowledge. Ford even acknowledges that each of Axling and Fohn “discusses all aspects of defining and configuring systems.” (Pet., p. 18.) Beyond this, Ford simply argues that the two references complement each other because their focuses differ. This is insufficient to sustain an obviousness combination.

Here, Ford’s failure to identify the differences between the asserted references and the claimed subject matter means that it has failed to make a proper obviousness analysis under 35 U.S.C. § 103(a). *Travelocity.com*, p. 20. As such, Ford fails to adequately state a claim for relief because “[a] petitioner has the burden of proof to establish that it is entitled to the requested relief.” *Liberty Mut.*

*Ins. Co. v. Progressive Cas. Ins. Co.*, CBM2012-00003, Paper No. 7, pp. 2, 3 (P.T.A.B. Oct. 25, 2012) (citing 37 C.F.R. § 42.20(c)).

The Board should, therefore, deny Ford's grounds for at least this additional reason.

**2. Ford fails to provide a reason to combine the applied references on an element-by-element basis.**

Ford correctly points out that Axling and Fohn are extremely similar. Both address the problem of the high complexity and limited usability of the original configurators, and both disparage previous systems as too burdensome and complex. But the result is two separate and independent systems that each independently purportedly solve a problem. Thus, a person having ordinary skill in the art using either one would have had no reason to implement details of the other. And Ford has not provided any evidence to the contrary short of pure hindsight. (Pet., pp.17-18.)

Moreover, Ford's failure to properly address claim construction results in a fundamental inability to apply the cited references to the claimed subject matter. In its petition, Ford indicates that it "does not believe any terms in the challenged claims require construction beyond their plain and ordinary meaning under the broadest reasonable interpretation standard required for this proceeding." (Pet., p. 16.) However, in a Decision entered July 14, 2016 (Paper No. 9), the Board



granted Versata's "Motion for District Court (*Phillips*) Claim Construction" (Paper No. 8). As a result, the petition is legally deficient because it lacks the requisite specificity to support a finding of obviousness, as Ford's analysis of the applied references is made using the incorrect standard. Ford did not oppose the Motion, nor did Ford seek to clarify the record with regard to the proper construction. This is despite Ford's seeming acknowledgement that additional claim construction briefing might be necessary if the claim construction standard is modified. (Pet., p. 16.) Regardless, it is not appropriate at this time for Ford to provide the necessary arguments without forfeiting the petition's filing date.

To establish obviousness, it is not enough to simply show that the various claim limitations are present in the prior art as a whole. Instead, it can be important to "identify a reason that would have prompted a person of ordinary skill in the relevant field to combine the elements in the way the claimed new invention does." *KSR Int'l Co.*, 550 U.S. at 418; *see also Black & Decker, Inc. v. Positec USA, Inc.*, 2016 WL 2898012, at \*5 (Fed. Cir. May 18, 2016) (reversing the Board when it failed to explain why a person of ordinary skill in the art would modify the teaching of one reference with another reference).

Ford has not done that here -- instead, Ford relies on relies generally upon the knowledge of a person of ordinary skill in the art to fill gaps in the art. The presence of these gaps, and Ford's failure to fill those with written prior art

references, demonstrates Ford's inability to find all elements of the claims described in the art prior to the priority date (or to find those elements in a manner that is combinable with the other written prior art references). And while Grounds 2 and 3 do not explicitly rely on the knowledge of a person of ordinary skill in the art, they apply to claims that depend from ones at issue in the first ground.

Ford should have identified why a person having ordinary skill in the art would combine the references to teach each of the claim features. Instead, Ford relied on impermissible hindsight. *See Graham*, 383 U.S. at 36 (U.S. 1966). The Board should, therefore, deny Ford's petition for this additional reason.

## V. CONCLUSION

Versata has demonstrated that Ford's proposed grounds are legally and factually deficient, and fail to show that claims 60-72 of the '651 patent are obvious over the cited references. Because Ford has failed to demonstrate a reasonable likelihood of prevailing on any proposed ground of invalidity, the Board should not institute trial in this *inter partes* review proceeding.

Date: August 12, 2016

Respectfully submitted,

By: 

Robert Greene Sterne (Reg. No. 28,912)  
Salvador M. Bezos (Reg. No. 60,889)  
Jonathan Tuminaro (Reg. No. 61,327)  
Joseph E. Mutschelknaus (Reg. No. 63,285)  
STERNE, KESSLER, GOLDSTEIN & FOX P.L.L.C.  
1100 New York Avenue, N.W.  
Washington, D.C. 20005-3934  
(202) 371-2600

Kent B. Chambers (Reg. No. 38,839)  
TERRILE, CANNATTI, CHAMBERS & HOLLAND, LLP  
11675 Jollyville Road, Suite 100  
Austin, TX 78759  
(512) 338-9100

Sharoon Saleem (Reg. No. 63,711)  
JONES & SPROSS, P.L.L.C.  
1605 Lakecliff Hills Ln., Suite 100  
Austin, TX 78732-2437  
(713) 446-0217

**CERTIFICATION OF WORD COUNT (37 C.F.R. § 42.24(b)(1))**

The undersigned hereby certifies that the portions of the above-captioned **PATENT OWNER'S PRELIMINARY RESPONSE** specified in 37 C.F.R. § 42.24 have 9,943 words, in compliance with the 14,000 word limit set forth in 37 C.F.R. § 42.24(b)(1). This word count was prepared using Microsoft Word 2010.

Respectfully submitted,

Date: August 12, 2016

By:   
Robert Greene Sterne, Registration No. 28,912  
**STERNE, KESSLER, GOLDSTEIN & FOX P.L.L.C.**  
1100 NEW YORK AVENUE, NW  
WASHINGTON, D.C. 20005  
(202) 371-2600

*Lead Counsel for Patent Owner  
Versata Development Group, Inc.*

**CERTIFICATE OF SERVICE (37 C.F.R. § 42.6(e))**

The undersigned hereby certifies that a true and correct copy of the enclosed **PATENT OWNER'S PRELIMINARY RESPONSE** was served electronically via e-mail on August 12, 2016, in its entirety on the following counsel of record for Petitioner:

John S. Leroy (Lead Counsel)  
Thomas A. Lewry (Back-up Counsel)  
Frank A. Angileri (Back-up Counsel)  
John P. Rondini (Back-up Counsel)  
Christopher C. Smith (Back-up Counsel)  
Jonathan D. Nikkila (Back-up Counsel)

BROOKS KUSHMAN P.C.  
1000 Town Center, Twenty-Second Floor  
Southfield, Michigan 48075

[FPGP0126IPR1@brookskushman.com](mailto:FPGP0126IPR1@brookskushman.com)

A handwritten signature in blue ink, reading "Robert Greene Sterne", is written over a horizontal line.

Robert Greene Sterne, Registration No. 28,912  
**STERNE, KESSLER, GOLDSTEIN & FOX P.L.L.C.**  
1100 NEW YORK AVENUE, NW  
WASHINGTON, D.C. 20005  
(202) 371-2600

***Lead Counsel for Patent Owner  
Versata Development Group, Inc.***