

EXHIBIT A-1 (SURESH)

Reference and Inventor	Country of Origin	Date of Publication	Title
U.S. Patent No. 6,208,990 to Suresh.	U.S.	Filed July 15, 1998	Method and architecture for automated optimization of ETL throughput in data warehousing applications

The above referenced U.S. Patent anticipates, or alternatively, renders obvious, the asserted claims of U.S. Patent No. 6,308,178 (the “’178 patent”).

To the extent that the following contentions reflect constructions of claim limitations consistent with or suggested by the Court and/or Plaintiff, no inference is intended nor should any be drawn that Defendants agree with the Court’s and/or Plaintiff’s claim constructions, and Defendants expressly reserve their rights to contest such claim constructions. Defendants offer such contentions solely in the alternative to any position it has taken or may ultimately take as to any claim construction issues.

The following contentions are subject to revision and amendment pursuant to Federal Rule of Civil Procedure 26(e) and the Orders of record in this matter subject to further investigation and discovery regarding the reference and the Court’s construction of the claims at issue.

Suresh was filed on July 15, 1998. Suresh is prior art to the ’178 patent under at least 35 U.S.C. § 102(a).

Citations to particular documents or passages are merely exemplary of where specifically in Suresh or other references each limitation of each asserted claim of the ’178 patent is found. Defendants reserve the right to rely on other documents or passages providing comparable evidence of how Suresh alone or in combination with other prior art, renders the asserted claim of the ’178 patent invalid. Further, Plaintiff’s infringement contentions are vague and ambiguous, do not comply with the requirements of this District’s Standing Order Governing Proceedings (OGP)

4.1 – Patent Cases, and Scheduling Order and do not provide adequate notice as to Plaintiff's infringement theories. Defendant also reserve the right to amend and supplement its contentions once Plaintiff has provided adequate notice of Plaintiff's infringement theories.

Claim Language of U.S. Patent No. 6,308,178	Suresh alone or in view of the prior art
1. A system for integrating data among heterogeneous source applications and destination applications, comprising:	Suresh discloses a system for integrating data among heterogeneous source applications and destination applications. “In the present embodiment of the present invention, these operational databases provide the data source from which the transformation engine server 240 can extract, transform, then load the data into the databases 250 and 260. Databases 250 and 260 are the data warehouses or data marts that are the targets of the data ETL [Extraction/Transformation/Loading] process.” 6:16-19 “A client is used to create a repository, which is used to keep track of session information as well as mapping information relating to how data is to be mapped and transformed from target tables of the operational databases to target tables of data marts.” 6:23-29 “The operational databases 210, 220, and 230 store the data resulting from business and financial transaction, and/or from equipment performance logs. These databases can be any of the conventional RDMS systems (such as from Oracle, Informix, Sybase, Microsoft, etc.) that reside within a high capacity mass storage device (such as hard disk drives, optical drives, tape drives, etc.).” 6:8-15 “In the present embodiment of the present invention, these operational databases provide the data source from which the transformation engine server 240 can extract, transform, then load the data into the databases 250 and 260. Databases 250 and 260 are the data warehouses or data marts that are the targets of the data ETL [Extraction/Transformation/Loading] process.” 6:15-20

**Claim Language of U.S.
Patent No. 6,308,178**

Suresh alone or in view of the prior art

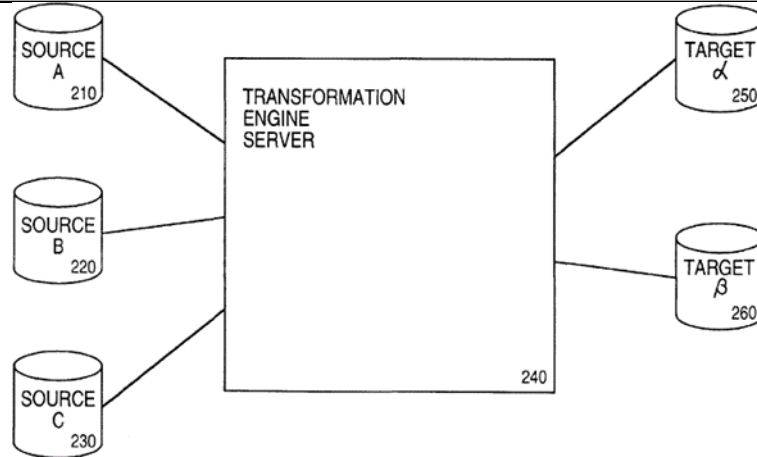


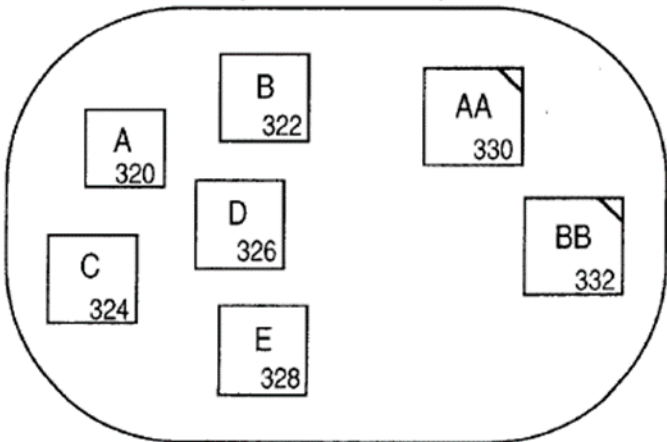
Figure 2

“Transformation engine is synonymous with a transformation server, which is a software server that hosts and oversees the data ETL process from the source database to the target databases. Source databases are operational databases from which raw data is extracted to be transformed into the transformation engine. Target databases are data warehouses and/or data marts into which transformed data is loaded.” 5:8-17

“With reference to FIG. 1, portions of the present invention are comprised of the computer-readable and computer executable instructions which reside, for example, in computer system 10 used as a part of a data warehousing system in accordance with one embodiment of the present invention.” 5:36-41

Claim Language of U.S. Patent No. 6,308,178	Suresh alone or in view of the prior art
	“But given the size and scope of the operational databases and given that there might exist numerous operational databases and many different types of data marts, this approach requires a monumental software development effort for incorporation, synchronizing, and updating the changes made to an operational databases so that they are appropriately reflected in the data warehouses and data marts.” 2:25-32 To the extent that that Suresh’s Sources and Targets are not applications, it would have been obvious at the time of the alleged invention to provide applications instead of databases or data warehouses. For example, Brann (Exhibit A-4) describes the problem of integration and communication between different types of computer programs, and provides a solution to that problem. Brann provides “an efficient system for assembling and for translating information from one computer to another.” Brann at 1:10-13. One of ordinary skill in the art would have understood that the databases or data warehouses of Suresh could be replaced with applications to solve the problem of interoperability between and integration of different types of applications. Furthermore, this claim and limitation would be obvious in light of Suresh alone or in combination with the references cited in the Obviousness Chart for at least the same reasons. See Exhibit A-20. It would have been obvious to a person of ordinary skill in the art presented with the teachings of Suresh to turn to the teachings of the other prior art to result in operation that would have been predictable in nature and would improve the performance of heterogeneous computer systems and applications.
a repository of information relating to the source applications and the destination applications, the repository including a plurality of data elements	Suresh discloses a repository of information relating to the source applications and the destination applications, the repository including a plurality of data elements each being capable of performing a discrete operation on a piece of data, and a first module including a plurality of instruction sets, each instruction of each instruction set activating a data element. “Data relating to operational databases can be entered, updated, and retrieved by clients. A client is used to create a repository, which is used to keep track of session information as

Claim Language of U.S. Patent No. 6,308,178	Suresh alone or in view of the prior art
each being capable of performing a discrete operation on a piece of data, and a first module including a plurality of instruction sets, each instruction of each instruction set activating a data element; and	well as mapping information relating to how data is to be mapped and transformed from target tables of the operational databases to target tables of data marts.” 6:23-29 “Instead of writing one block of source code that is responsible for all phases of the data ETL process, a more disciplined and architectural approach is taken in packing the source code such that the responsibility is distribute to smaller units (i.e., components) of source code. Each one of these software components is responsible for one specific type of transformation.” 6:46-52 “In FIG. 3C, an extensible set 350 of transformation components is formed by transformation components 320, 322, 324, 326, and 328 which were originated from a monolithic transformation application such as M 310, together with transformation components developed by the users (e.g., blocks 330 and 332). This extensible set 350 of transformation components can be thought of as a pool (i.e., a base of ready-made transformation components from which the users can draw on to build functionally more sophisticated transformations in the data ETL process).” 6:56-65 “FIG. 5 illustrates the manual selection 515 of transformation components that are being formed into the pipeline 540. Basically, a set of transformations are componentized so that they may be coupled together (e.g., in a Lego-like fashion).” 7:38-42

Claim Language of U.S. Patent No. 6,308,178	Suresh alone or in view of the prior art
	Figure 3C  350 Extensible Set of Transformation Components The diagram shows a large rounded rectangle labeled 350, which is identified as the 'Extensible Set of Transformation Components'. Inside this rectangle are six smaller boxes, each representing a transformation component. These are labeled as follows: 'A 320' (top left), 'B 322' (top center), 'AA 330' (top right), 'C 324' (middle left), 'D 326' (center), and 'E 328' (bottom center). Additionally, there is a box labeled 'BB 332' located outside the rounded rectangle to the right. Two arrows point from the header 'Suresh alone or in view of the prior art' down towards the top of the rounded rectangle 350.

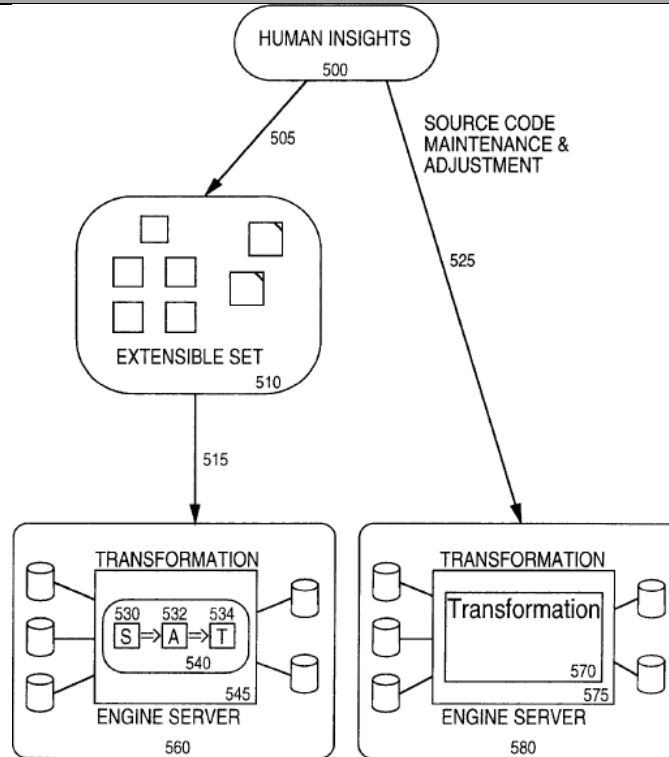


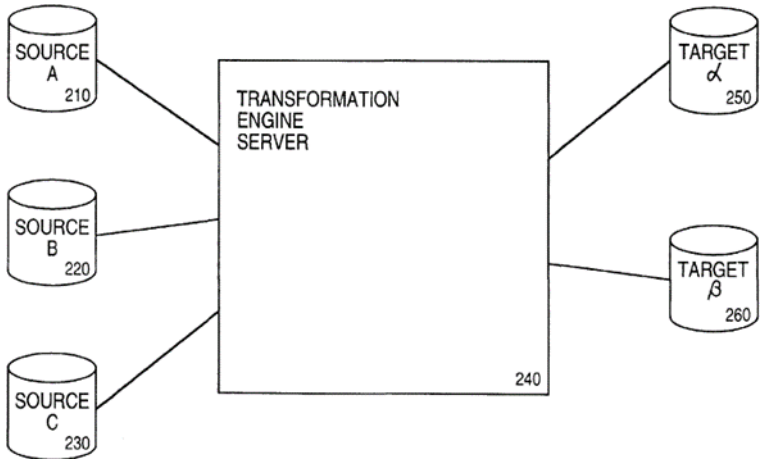
Figure 5

“In the currently preferred embodiment, there are thirteen different transformation components: source, target, expression, aggregation, filter, rank, update strategy, sequence, joiner, lookup, stored procedure, external procedure, and normalizer. The source

Claim Language of U.S. Patent No. 6,308,178	Suresh alone or in view of the prior art
	transformation contains tables, views, synonyms, or flat files that provide data for the data mart/data warehouse. The target transformation maintains database objects or files that receive data from other transformations. These targets then make the data available to data mart users for decision support. Expression transformations calculate a single result, using values from one or more ports. The aggregation transformation calculates an aggregate value, such as a sum or average, using the entire range of data within a port or within a particular group. Filter transformations filter (selects) records based on a condition the user has set in the expression. The rank transformation filters the top or bottom range of records, based on a condition set by the user. The update strategy transformation assigns a numeric code to each record indicating whether the server should use the information in the record to insert, delete, or update the target. The sequence generator transformation generates unique ID numbers. The joiner transformation joins records from different databases or file systems. The lookup transformation looks up values. The stored procedure transformation calls a stored procedure. The external procedure transformation calls a procedure in a shared library or in the COM layer of Windows NT. And the normalizer transformation normalizes records, including those read from virtual storage access method (VSAM) sources. In the currently preferred embodiment, the source, target, aggregation, rank, and joiner transformations are all staged transformations. The data generated by these transformations are automatically staged by the software, without human interaction. The expression, filter, update strategy, sequence, lookup, stored procedure, external procedure, and normalizer transformations are all streamed transformations. Other new types of transformations can be added to this list. These new transformations can be specified as being either staged or streamed.” 10:35-11:6. “A transformation component is the same as a transformation object, which is a software binary file acting as an individual unit that possesses built-in autonomy and encapsulates the functionalities of a transformation.” 5:17-20.

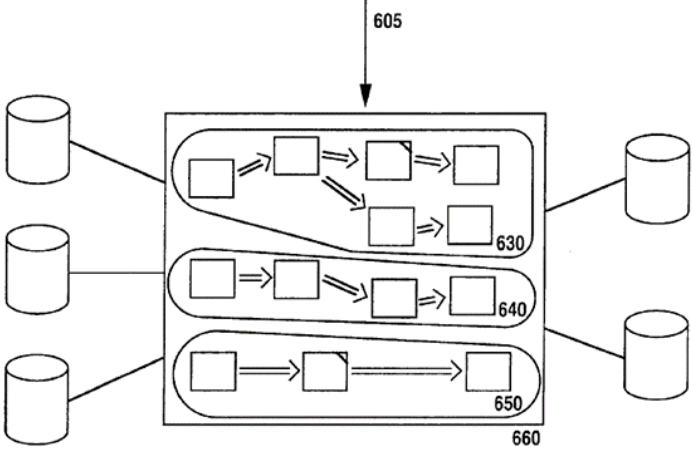
Claim Language of U.S. Patent No. 6,308,178	Suresh alone or in view of the prior art
	To the extent that this claim element is not disclosed by Suresh, it would be obvious to one or ordinary skill in the art to provide a first module including a plurality of instruction sets, each instruction of each instruction set activating a data element. For example, Brann (Exhibit A-4) describes a repository (database) includes data structures, for example, a first module (database table(s)) that store instruction sets for activating data structures (subroutines). “The late binding programs described consist of two parts-a Control Program and subroutines, which carry out the required program steps. The purpose of the Control Program is to receive the incoming stimuli, with their associated data, and derive the appropriate action to be taken. The action takes the form of a list of subroutines to be invoked and an order in which they should be invoked. The Control Program then invokes the required subroutines in the required order, checking that each invocation was successful. In the preferred embodiment of this invention, the programs may be written in any language; preferably the C programming language. The Control Program extracts its instructions from the database and retains them, in the form of arrays of data structures, in the computer's memory.” Brann at 7:21-35. The database stores a set of transformation instructions which consists of the originating data elements (i.e., a field in the first structure), a resulting data element (i.e., a field in the Second structure), and the Transformation Method required between them. Each one of those makes up one row in a table of the database. <i>Id.</i> at 10:58 – 11:10. A method table in the database stores a list of subroutines to be invoked and the order in which they should be invoked. <i>Id.</i> at 7:23-39; 11:10-23. Each instruction activates its associated subroutine. <i>Id.</i> at 7:21-35. The method table includes pointers to the particular subroutines. A function reference is used to activate the selected subroutine. <i>Id.</i> at 7:31-47; See also 11:10-19. It would be obvious to prior to provide a first module including a plurality of instruction sets, each instruction of each instruction set activating a data element, as it would be use of a known

Claim Language of U.S. Patent No. 6,308,178	Suresh alone or in view of the prior art
	technique to improve known devices or methods in the same way to yield predictable results; it would also be "obvious to try" because it would involve choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success. It would have been obvious to do so to store and combine transformation components at run time. Brann at 3:9-15. It would have been obvious to do so to eliminate the need for additional programming and custom software; to reduce users' learning curves; to eliminate manual data entry; and to allow for upgrades to the applications. Brann at 2:45-62; 7:63 – 8:4. Updates may be accomplished by revising the extensible set to add new transformation components and by adding new instructions to the database for different transformations. Suresh at 11:4-6. Furthermore, this claim and limitation would be obvious in light of Suresh alone or in combination with the references cited in the Obviousness Chart for at least the same reasons. See Exhibit A-20. It would have been obvious to a person of ordinary skill in the art presented with the teachings of Suresh to turn to the teachings of the other prior art to result in operation that would have been predictable in nature and would improve the performance of heterogeneous computer systems and applications.
an active component connected to the repository having an input for loading data from the source applications, and an output for populating the destination applications with processed data, the active component	Suresh discloses a system for integrating data among heterogeneous source applications and destination applications. “In the present embodiment of the present invention, these operational databases provide the data source from which the transformation engine server 240 can extract, transform, then load the data into the databases 250 and 260. Databases 250 and 260 are the data warehouses or data marts that are the targets of the data ETL [Extraction/Transformation/Loading] process.” 6:16-19. “A client is used to create a repository, which is used to keep track of session information as well as mapping information relating to how data is to be mapped and transformed from

Claim Language of U.S. Patent No. 6,308,178	Suresh alone or in view of the prior art
processing the data from the source applications by activating data elements according to the plurality of instruction sets in the module.	target tables of the operational databases to target tables of data marts.” 6:23-29. “The operational databases 210, 220, and 230 store the data resulting from business and financial transaction, and/or from equipment performance logs. These databases can be any of the conventional RDMS systems (such as from Oracle, Informix, Sybase, Microsoft, etc.) that reside within a high capacity mass storage device (such as hard disk drives, optical drives, tape drives, etc.).” 6:8-15. “In the present embodiment of the present invention, these operational databases provide the data source from which the transformation engine server 240 can extract, transform, then load the data into the databases 250 and 260. Databases 250 and 260 are the data warehouses or data marts that are the targets of the data ETL [Extraction/Transformation/Loading] process.” 6:15-20.  Figure 2

Claim Language of U.S. Patent No. 6,308,178	Suresh alone or in view of the prior art
	“Transformation engine is synonymous with a transformation server, which is a software server that hosts and oversees the data ETL process from the source database to the target databases. Source databases are operational databases from which raw data is extracted to be transformed into the transformation engine. Target databases are data warehouses and/or data marts into which transformed data is loaded.”5:8-17 “With reference to FIG. 1, portions of the present invention are comprised of the computer-readable and computer executable instructions which reside, for example, in computer system 10 used as a part of a data warehousing system in accordance with one embodiment of the present invention.” 5:36-41 “But given the size and scope of the operational databases and given that there might exist numerous operational databases and many different types of data marts, this approach requires a monumental software development effort for incorporation, synchronizing, and updating the changes made to an operational databases so that they are appropriately reflected in the data warehouses and data marts.” 2:25-32 Furthermore, this claim and limitation would be obvious in light of Suresh alone or in combination with the references cited in the Obviousness Chart for at least the same reasons. See Exhibit A-20. It would have been obvious to a person of ordinary skill in the art presented with the teachings of Suresh to turn to the teachings of the other prior art to result in operation that would have been predictable in nature and would improve the performance of heterogeneous computer systems and applications.
3. The system of claim 1 further comprising a second module including	Suresh discloses the system of claim 1 further comprising a second module including a plurality of instruction sets, each instruction of each instruction set activating a data element, the first module plurality of instruction sets being defined for interface with a first group of destination

Claim Language of U.S. Patent No. 6,308,178	Suresh alone or in view of the prior art
a plurality of instruction sets, each instruction of each instruction set activating a data element, the first module plurality of instruction sets being defined for interface with a first group of destination applications, the second module plurality of instruction sets being defined for interface with a second group of destination applications.	applications, the second module plurality of instruction sets being defined for interface with a second group of destination applications. See claim 1. “FIG. 6 illustrates the pipeline-based aspect of the architecture and the level of pipelines of transformation components. An exemplary arrangement of transformation components that the users have selected from the extensible set of transformation components is shown.” 7:66-8:3 “Transformation system 600 shows that three independent pipelines have naturally emerged. Each of the separate processes 630, 640, and 650 marks (separates) out the natural boundary of an independent task to be executed by the transformation engine server.” 8:5-9 “A single transformation can accept one or more sets of inputs from one or more sources and/or prior transformations and can output one or more data sets to one or more subsequent transformations and/or targets.” 8:31-35 “The pipelined transformations exist within the transformation engine server 460. Data from any of the source tables 450, 452, and 454 can be transformed by the pipelined transformations within transformation engine server 460 and stored in either and/or both of target tables 470 and 472. Different transformations can be added; existing transformations can be removed; designated transformations can be moved to a different portion of the pipeline; additional pipeline stages can be created; multiple processes can run simultaneously within parallel pipeline stages; etc..” 7:27-37

Claim Language of U.S. Patent No. 6,308,178	Suresh alone or in view of the prior art
	 Figure 6

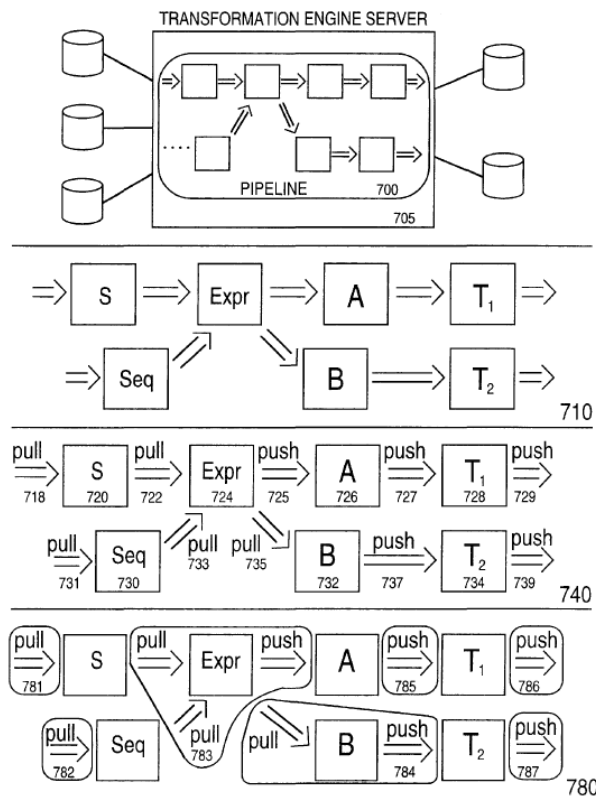
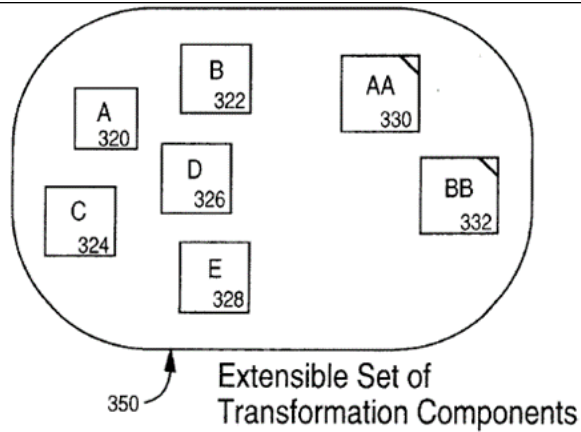


Figure 7

Furthermore, this claim and limitation would be obvious in light of Suresh alone or in combination with the references cited in the Obviousness Chart for at least the same reasons. See Exhibit A-20 It would have been obvious to a person of ordinary skill in the art presented with the teachings of Suresh to turn to the teachings of the other prior art to result in operation

Claim Language of U.S. Patent No. 6,308,178	Suresh alone or in view of the prior art
	that would have been predictable in nature and would improve the performance of heterogeneous computer systems and applications. For example, Suresh discloses that data from source tables can be stored in different target tables. Suresh at 8:31-35. Suresh teaches the cascading sequences in the first pipeline that interface with one target and the cascading sequences in the second pipeline that interface with two targets, which renders obvious “the first module plurality of instruction sets being defined for interface with a first group of destination applications, the second module plurality of instruction sets being defined for interface with a second group of destination applications,” as claimed.
4. The system of claim 3 wherein at least one data element of the plurality of data elements is activated by both the first module plurality of instruction sets and the second module plurality of instruction sets.	Suresh discloses the system of claim 3 wherein at least one data element of the plurality of data elements is activated by both the first module plurality of instruction sets and the second module plurality of instruction sets. See claim 3. “In FIG. 3C, an extensible set 350 of transformation components is formed by transformation components 320, 322, 324, 326, and 328 which were originated from a monolithic transformation application such as M 310, together with transformation components developed by the users (e.g., blocks 330 and 332). This extensible set 350 of transformation components can be thought of as a pool (i.e., a base of ready-made transformation components from which the users can draw on to build functionally more sophisticated transformations in the data ETL process).” 6:56-65

Figure 3C



“The pipelined transformations exist within the transformation engine server 460. Data from any of the source tables 450, 452, and 454 can be transformed by the pipelined transformations within transformation engine server 460 and stored in either and/or both of target tables 470 and 472. Different transformations can be added; existing transformations can be removed; designated transformations can be moved to a different portion of the pipeline; additional pipeline stages can be created; multiple processes can run simultaneously within parallel pipeline stages; etc..” 7:27-37

“Transformation system 600 shows that three independent pipelines have naturally emerged. Each of the separate processes 630, 640, and 650 marks (separates) out the natural boundary of an independent task to be executed by the transformation engine server.” 8:5-9

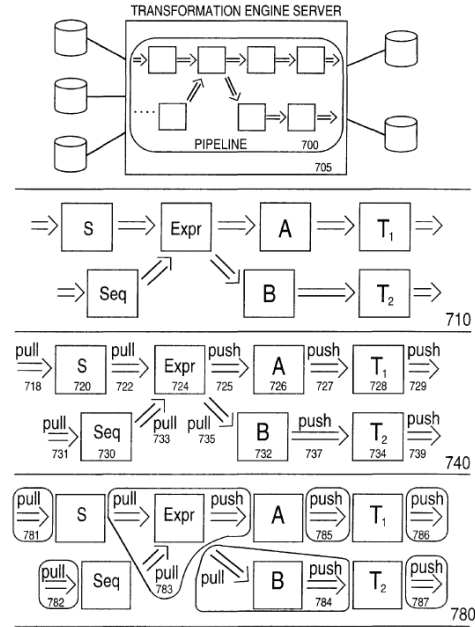


Figure 7

“In the currently preferred embodiment, there are thirteen different transformation components: source, target, expression, aggregation, filter, rank, update strategy, sequence, joiner, lookup, stored procedure, external procedure, and normalizer.” 10:35-37

Furthermore, claim 4 would have been obvious in view of Suresh alone or in combination with other prior art references, including, without limitation, one or more references identified in Defendants’ Invalidity Contentions.

7. The system of claim 1 wherein the repository is capable of receiving a plurality of modules.

Suresh discloses the system of claim 1 wherein the repository is capable of receiving a plurality of modules.

See claim 1.

“A client is used to create a repository, which is used to keep track of session information as well as mapping information relating to how data is to be mapped and transformed from target

	tables of the operational databases to target tables of data marts.” 6:25-29 “Again, the extensible set 400 of transformation components is shown here as a repository of transformation components available for use to build functionally more sophisticated transformations.” 7:5-7:8 “Arrow 410 indicates the step taken by the users and enabled by the componentized aspect of the architecture to manually form a new transformation that consists of a cascading sequence of transformation components.” 7:8-11 “Transformation system 600 shows that three independent pipelines have naturally emerged. Each of the separate processes 630, 640, and 650 marks (separates) out the natural boundary of an independent task to be executed by the transformation engine server.” 8:5-9 Furthermore, this claim and limitation would be obvious in light of Suresh alone or in combination with the references cited in the Obviousness Chart for at least the same reasons. See Exhibit A-20. It would have been obvious to a person of ordinary skill in the art presented with the teachings of Suresh to turn to the teachings of the other prior art to result in operation that would have been predictable in nature and would improve the performance of heterogeneous computer systems and applications. For example, in such a modification, the transformation engine server would have easy access to the pipelines when the system administrator requests or schedules the pipeline execution. Further, Suresh discloses that the repository stores mapping information, and the pipelines are the mapping information that describes how the data is transformed using the cascading sequence of the transformation components. See Suresh, 6:25-29, 7:9- 11. Accordingly, because the pipelines are mapping information, one of ordinary skill in the art would have recognized that the user-made new pipelines would be stored in the repository.
8. The system of claim 1 wherein the destination applications are included within an ERP product.	See claim 1. Furthermore, this claim and limitation would be obvious in light of Suresh alone or in combination with the references cited in the Obviousness Chart for at least the same reasons. See Exhibit A-20. It would have been obvious to a person of ordinary skill in the art presented

	with the teachings of Suresh to turn to the teachings of the other prior art to result in operation that would have been predictable in nature and would improve the performance of heterogeneous computer systems and applications. For example, Beauchamp teaches that it was desirable for data integration systems to integrate all applications, including ERP systems. Beauchamp at 19:31-34. It would be obvious to so so the ERP system would be able to access and use the data transformed and loaded from other databases applications
--	---