

EXHIBIT A-4 (BRANN)

Reference and Inventor	Country of Origin	Date of Publication	Title
U.S. Patent No. 5,842,205 to Brann	U.S.	November 24, 1998	Method and process of inter-machine communication and generalized method for program preparation therefor

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.

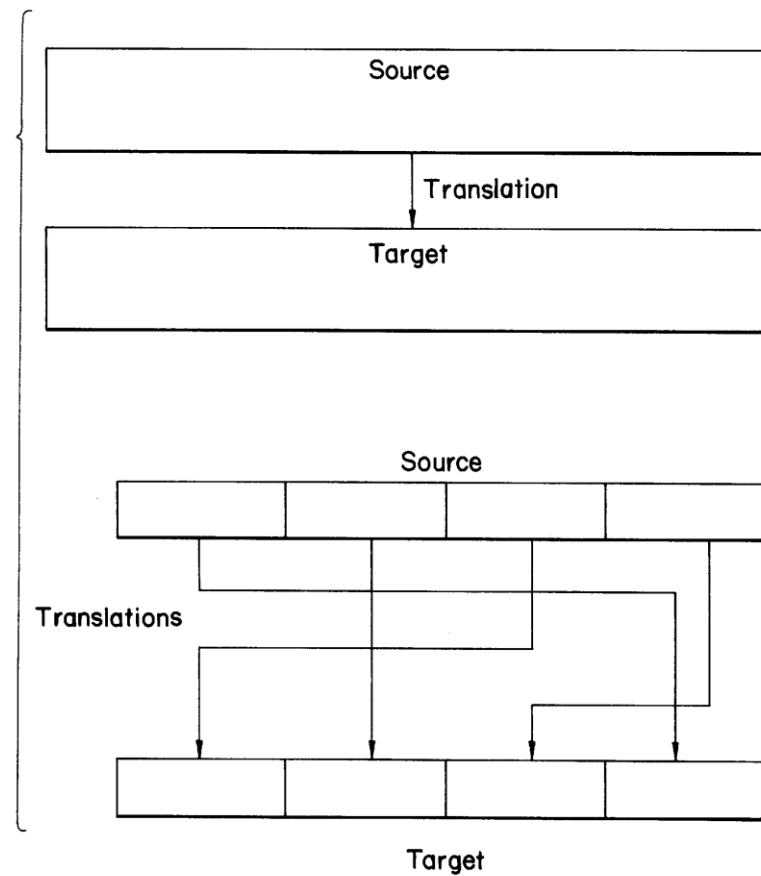
Brann was published on November 24, 1998, Brann is prior art to the ’178 patent under at least 35 U.S.C. § 102(a).

Citations to particular documents or passages are merely exemplary of where specifically in Brann 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 Brann 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	Brann alone or in view of the prior art
1. A system for integrating data among heterogeneous source applications and destination applications, comprising:	Brann discloses a system for integrating data among heterogeneous source applications and destination applications. “More particularly, this invention relates to a late binding process for providing an efficient system for assembling and for translating information from one computer to another.” 1:10-13 “The history of computer systems and the programs used therein have developed quickly over the past thirty five years. The earliest processes to be automated were large scale, batch data processing tasks, Such as payroll production. Thereafter, other problems were addressed. As each new problem is addressed and solution implemented, an additional body of computer programs is created. The nature of the industry is such, however, that new programs are preferred over improvements to existing solutions. This has resulted in new systems being built which are unrelated to those which already exist. Over time, the number of independent systems has grown enormously. There has been little attention paid to coordination of the data between the various competing programs.” 1:16-30 “The overall problems in replacing systems caused by the logistic of converting the data contained in such systems, implementing the new software, and re-education of users means that such systems will continue to exist for a considerable period of time. This means that these systems must be used in multi-user, networked systems so that the data these systems contain can be used by later-developed system. Yet another difficulty lies in the interaction between different systems.” 2:45-55

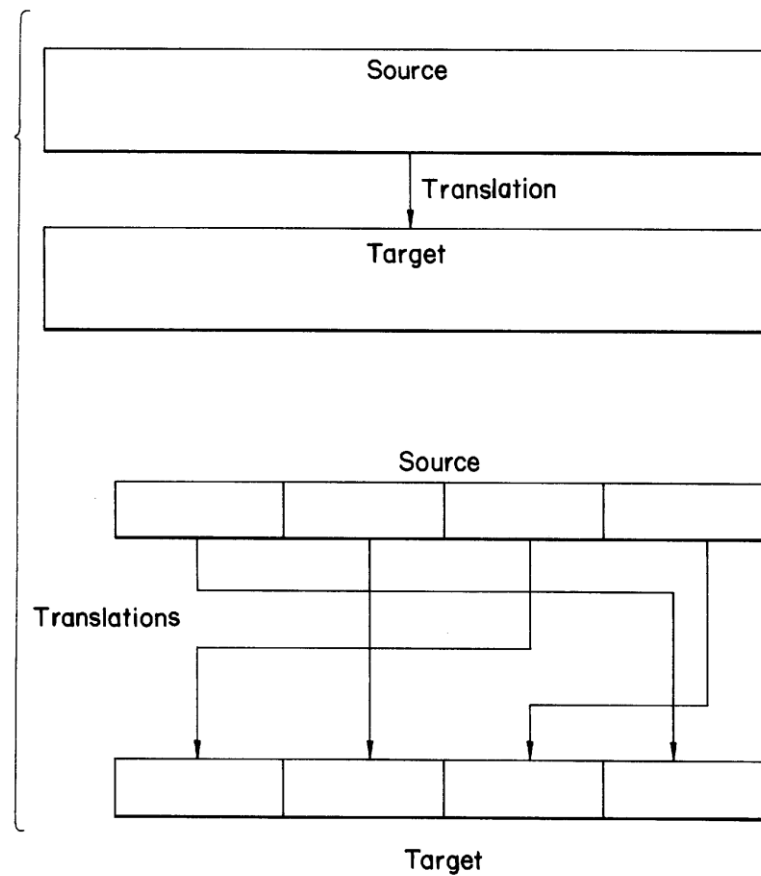
FIG. 4



“The third task acknowledges that different computer systems represent information in different forms.” 7:8-10

Claim Language of U.S. Patent No. 6,308,178	Brann alone or in view of the prior art
	“There is a need for a mechanism which will aid in the rapid integration of computer systems and a second need for a faster way of developing and changing systems. A single solution is capable of providing an answer to both of these problems.” 5:27-31 “The Control Program also looks up the size of the third field in the Metadatabase.” 11:46-47
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; and	Brann describes 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. “Storing instructions, data, and the structures of any source stimuli to be acted upon in the database.” 3:49-51 “The database contains references to the processing steps to be carried out for a particular stimulus.” 6:49-50

FIG. 4



“The result of this analysis is 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

Claim Language of U.S. Patent No. 6,308,178	Brann alone or in view of the prior art
	in the Second structure), and the Transformation Method required between them. Each one of those makes up one row in the XLAT details table of the Metadatabase. This may be set aside in the translation details table as XLAT-DETAILS. The data is then entered into tables in the Metadatabase.” 10:58-66. “For instance, one of the instructions might be "copy the third field in the incoming to the sixth field in the outgoing”. The Control Program locates the third field in terms of byte addresses. The Control Program also looks up the size of the third field in the Metadatabase.” 11:43-47. “Accessing the database to retrieve information relating to a type of the provided stimulus message.” 14:47-48. “The set of subroutines required to implement a Translation Program consists, therefore, of a set of structural conversion routines. To carry out functions like: Copy (with justification to left or right, as appropriate) Convert from character to integer numeric formats (and vice versa) Convert from character to floating point numeric formats (and vice versa) Convert from floating point to integer numeric formats (and vice versa) Lookup incoming data file in table of values (output result as characters, integer, or floating point) A further list of representative functions is attached in the appendix. When invoked, each of these routines is passed a reference to the data it must take as input, a reference to the place its output must be written to, the maximum allowable lengths of input and output, and a reference to the lookup table (if any) being employed.” 8:36-54

Claim Language of U.S. Patent No. 6,308,178	Brann alone or in view of the prior art
	“The nature of the instructions is not defined, but are generally taken to be an atomic unit of processing appropriate to the task in hand. In fact, in the construction of computer systems, the form of the “instructions” is usually defined by the task at hand and is then expressed using a suitable programming language. It is well known that such functions are best implemented as subroutines which can be re-used in similar instruction needs” 6:29-36. “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.” 7:22-30. “These subroutines are capable of being assembled into any conceivable Translation Program.” 9:13-14 “The translational steps are described in a database in which all necessary instructions and data are stored.” Abstract. “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. (This extraction of data structures into memory structures is not essential, but it does enable the program to perform very much faster.) One of the data arrays consists of all of the required series of subroutine invocations. These data structures contain an executable function

Claim Language of U.S. Patent No. 6,308,178	Brann alone or in view of the prior art
	reference (in the C language this is a "function pointer") to the subroutine which is the reference required by the program in order to execute the subroutine. This function pointer cannot be held in the database. In order to determine the function pointers to all the required subroutines, the database contains a unique string of characters for each available subroutine (this is the function name and is actually coded into each subroutine). The Control Program contains a function which can convert these function names into function pointers.” 7:31-49. “Translation Programs employ Translation Methods for all their translation activities. In order to extend the processing model of Control Program and subroutines to encompass general purpose programs, it is necessary to extend the range of subroutines which can be executed. It is particularly valuable to be able to use subroutines which were not purpose built for this environment. The ability to re-use well established code constructed for use in other programs enables a program using the Control Program and subroutine architecture to be constructed extremely rapidly for new applications.” 9:30-40 “The data is then entered into tables in the Metadatabase. The existence of a translator is recorded in the location table, also recorded here is the Identification Method to be invoked to distinguish between the incoming messages and the section of the incoming messages to be examined by the Identification Method. Each of the incoming message types is recorded in the source table together with any arbitrary data to be found in messages of this type in the section defined in the location table. Each of the target messages for each source message is recorded in the target table. Each of the details of the transformation is recorded in a translation details table.” 10:58-11:10 “For each transformation there is a Method that is invoked (i.e., a subroutine that carries out the particular transformation required) which is represented by a reference back to the methods table in the Metadatabase. The methods table itself contains a name for an executable subroutine (i.e., a Method). That name is used by the Control Program to derive the executable function reference of the appropriate subroutine when it reads the database. The final result for each of the transformations is identified as it comes to this reference. It then resolves the

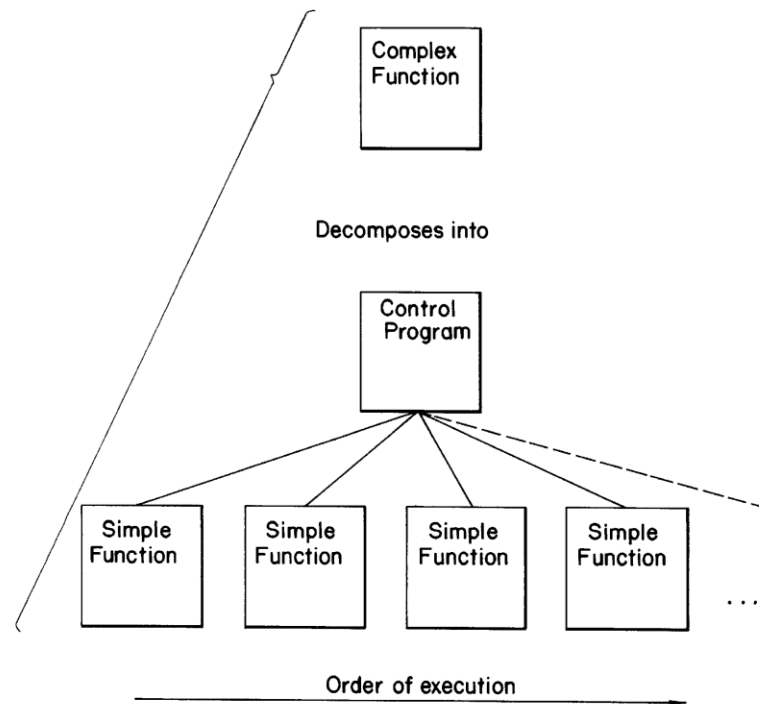
Claim Language of U.S. Patent No. 6,308,178	Brann alone or in view of the prior art
	reference to find the subroutine and execute the subroutine. The other information that it has to pass to the subroutine, which is parameter information, consists of a pointer to where the source field is in its temporary computer memory; where the target field is in its temporary computer memory; the size of those fields in bytes, and, if necessary, a reference to a look up table which would be used in nonalgorithmic transformations.” 11:10-26 “Translation Methods are written as subroutines which receive a particular set of parameters. Each Method carries out a particular type of transformation. For example, there is one to do copying and one to do conversion of character format numbers into integers. Each of the Methods carries out a particular type of manipulation of simple data types. The number of simple data types is very small. As a consequence, there is provided a complete set of such Methods.” 12:47-55 “The Control Program is a program which invokes basic computing steps in the form of Methods. It calls upon databases to tell it or instruct it as to which of these Methods should be used under which circumstances of incoming data. The basic steps are created outside the Control Program and new ones can always be added without changing the Control Program. Therefore, any changes made to the program are to be made in the Methods or the databases which the Methods use.” 14:20-28 To the extent that this claim element is not disclosed by Brann, it would be obvious to one or ordinary skill in the art to provide a repository including a plurality of data elements, such as in Suresh, in Brann. Also, Suresh describes storing the data elements (“transformation components”) in a repository. Suresh at Fig 3. It would be obvious to prior to do so as it would be use of a known 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.

Claim Language of U.S. Patent No. 6,308,178	Brann alone or in view of the prior art
	Furthermore, this claim and limitation would be obvious in light of Brann alone or in combination with one or more of 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 Brann 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 processing the data from the source applications by activating data elements according to the plurality of instruction sets in the module.	Brann describes 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 processing the data from the source applications by activating data elements according to the plurality of instruction sets in the module. “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.” 7:22-30. “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. (This extraction of data structures into memory structures is not essential, but it does enable the program to perform very much faster.) One of the data arrays consists of all of the required series of subroutine invocations. These data structures contain an executable function reference (in the C language this is a "function pointer") to the subroutine which is the reference required by the program in order to execute the subroutine. This function pointer cannot be held in the database. In order to determine the function pointers to all the required subroutines, the database contains a unique string of characters for each available subroutine (this is the function name and is actually coded into each subroutine). The Control Program contains a function which can convert these function names into function pointers.” 7:31-49.

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

Brann alone or in view of the prior art

FIG. 3



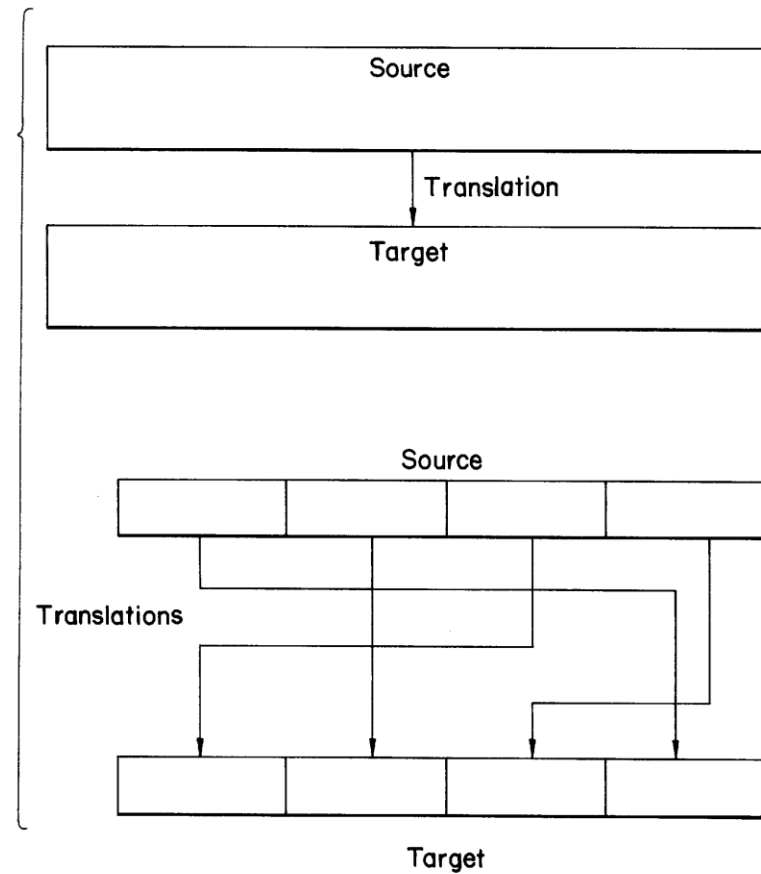
“The Control Program exists in a world where it is receiving stimuli. In effect it is being given incoming data to operate upon.

Claim Language of U.S. Patent No. 6,308,178	Brann alone or in view of the prior art
	Thus far, the Control Program is written to extract the data from the Metadatabase and derive the instructions from this data in terms of which Methods should be called under which stimuli. It is a mandatory requirement that there is nothing in the Control Program which refers to any other activity. The Control Program reads the Metadatabase and uses instructions found there to manipulate data structures that it is given.” 11:31-41 “Each time a Method executes, the executed Method tells the Control Program whether or not it successfully completed that operational step. The Control Program examines this returned value to ensure that each Method was successful, thereby checking that the source to target transformation has been successfully completed. When the Control Program has executed all the Methods it is directed to do, it delivers the outgoing structure to the receiver. The Control Program then waits for the next incoming signal or stimulus.” 12:1-10 “The control program, responding to the database and the translation method programs, then delivers the completely translated target message to the receiver.” Abstract “The Translation Program is, therefore, capable of carrying out any conceivable data translation, within the parameters of data structures, which can be described by the C programming language, which is capable of describing any data structure which can be represented on a modern digital computer.” 8:55-60 “The Control Program is a program which invokes basic computing steps in the form of Methods. It calls upon databases to tell it or instruct it as to which of these Methods should be used under which circumstances of incoming data. The basic steps are created outside the Control Program and new ones can always be added without changing the Control Program. Therefore, any changes made to the program are to be made in the Methods or the databases which the Methods use.” 14:20-28. Furthermore, this claim and limitation would be obvious in light of Brann alone or in combination with one or more of the references cited in the Obviousness Chart for at least the

Claim Language of U.S. Patent No. 6,308,178	Brann alone or in view of the prior art
	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 Brann 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 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.	Brann describes 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 applications, the second module plurality of instruction sets being defined for interface with a second group of destination applications. See claim 1. “The data is then entered into tables in the Metadatabase. The existence of a translator is recorded in the location table, also recorded here is the Identification Method to be invoked to distinguish between the incoming messages and the section of the incoming messages to be examined by the Identification Method. Each of the incoming message types is recorded in the source table together with any arbitrary data to be found in messages of this type in the section defined in the location table. Each of the target messages for each source message is recorded in the target table. Each of the details of the transformation is recorded in a translation details table.” 10:58-11:10 “For each transformation there is a Method that is invoked (i.e., a subroutine that carries out the particular transformation required) which is represented by a reference back to the methods table in the Metadatabase. The methods table itself contains a name for an executable subroutine (i.e., a Method). That name is used by the Control Program to derive the executable function reference of the appropriate subroutine when it reads the database. The final result for each of the transformations is identified as it comes to this reference. It then resolves the reference to find the subroutine and execute the subroutine. The other information that it has to

pass to the subroutine, which is parameter information, consists of a pointer to where the source field is in its temporary computer memory; where the target field is in its temporary computer memory; the size of those fields in bytes, and, if necessary, a reference to a look up table which would be used in nonalgebraic transformations.” 11:10-26

FIG.4



	Furthermore, this claim and limitation would be obvious in light of Brann alone in combination with one or more of 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 Brann 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.
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.	Brann describes that 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. “The nature of the instructions is not defined, but are generally taken to be an atomic unit of processing appropriate to the task in hand. In fact, in the construction of computer systems, the form of the “instructions” is usually defined by the task at hand and is then expressed using a suitable programming language. It is well known that such functions are best implemented as subroutines which can be re-used in similar instruction needs.” 6:29-36. “These subroutines are capable of being assembled into any conceivable Translation Program.” 9:13-14 “Translation Programs employ Translation Methods for all their translation activities. In order to extend the processing model of Control Program and subroutines to encompass general purpose programs, it is necessary to extend the range of subroutines which can be executed. It is particularly valuable to be able to use subroutines which were not purpose built for this environment. The ability to re-use well established code constructed for use in other programs enables a program using the Control Program and subroutine architecture to be constructed extremely rapidly for new applications.” 9:30-40 Furthermore, this claim and limitation would be obvious in light of Brann alone in combination with one or more of 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 Brann 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.
7. The system of claim 1 wherein the repository is capable of receiving a plurality of modules.	Brann describes the repository is capable of receiving a plurality of modules. See claim 1. “The result of this analysis is 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 the XLAT details table of the Metadatabase. This may be set aside in the translation details table as XLAT-DETAILS. The data is then entered into tables in the Metadatabase. The existence of a translator is recorded in the location table, also recorded here is the Identification Method to be invoked to distinguish between the incoming messages and the Section of the incoming messages to be examined by the Identification Method. Each of the incoming message types is recorded in the Source table together with any arbitrary data to be found in messages of this type in the Section defined in the location table. Each of the target messages for each Source message is recorded in the target table. Each of the details of the transformation is recorded in a translation details table.” 10:58 – 11:10. “In other words, the process of analysis populates the database in terms of the shapes and Structures of messages and which ones need to be translated and the Control Program interprets those as instructions for carrying out the translation.” 11:25-30.
8. The system of claim 1 wherein the destination applications are included within an ERP product.	Brann describes the destination applications are included within an ERP product. See claim 1. Furthermore, this claim and limitation would be obvious in light of Brann alone or in combination with one or more of 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 Brann 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. 19:31-34. It would be obvious to do so so that the ERP system would be able to access and use the data transformed and loaded from other databases applications.
--	--