

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
31 October 2002 (31.10.2002)

PCT

(10) International Publication Number
WO 02/086629 A2

- (51) International Patent Classification⁷: **G05B**
- (21) International Application Number: PCT/IE02/00055
- (22) International Filing Date: 25 April 2002 (25.04.2002)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
2001/0411 25 April 2001 (25.04.2001) IE
60/285,994 25 April 2001 (25.04.2001) US
- (71) Applicant (*for all designated States except US*): **FAST-NET SOFTWARE LIMITED** [IE/IE]; Cork Business & Technology Park, Model Farm Road, Cork (IE).
- (72) Inventors; and
- (75) Inventors/Applicants (*for US only*): **DEASY, Donal** [IE/IE]; 1 Carrigdena, Ballincollig, Cork (IE). **O'MAHONY, Eamonn** [IE/IE]; 15 Oakdene, Skehard Road, Blackrock, Cork (IE). **THOMPSON, John** [GB/IE]; 20 Ashford Court, Grange, Douglas, Cork (IE).
- (74) Agents: **O'BRIEN, John, A.** et al.; c/o John A. O'Brien & Associates, Third Floor, Duncairn House, 14 Carysfort Avenue, Blackrock, County Dublin (IE).
- (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: A PRODUCTION CONTROL TRACEABILITY SYSTEM AND METHOD

(57) Abstract: A traceability system (1) interfaces with discrete domain food production systems (3-8) and with continuous (production) domain production systems (9-12). Production data is uploaded to the traceability system (1) either by asynchronous transmission from the systems or by polling. Received production data is validated with reference to a production model. The model has objects (30) representing production stages and methods (40) which link the objects. The production data is used to record events in a database (60), each event having a source object identifier and a target object identifier. Subsequently, a trace is generated by retrieving trace events according to the linkages.



WO 02/086629 A2

- 1 -

"A production control and traceability system and method"INTRODUCTION5 Field of the Invention

The invention relates to production of products in the Life Sciences Industries, particularly products from bulk raw material such as dairy, food, beverage, pharmaceutical, and healthcare products.

10

Prior Art Discussion

Heretofore, much work has been carried out in automation and data capture of various aspects of such production. For example, in the dairy products industry
15 there is automatic data capture of data at the primary production stages, as follows:

at the farm an automatic data capture unit captures date, time, temperature, farm identity, and milk volume,

20 a collection sample from the farm provides percentage fat, protein, chemical and micro analysis data,

at the milk processing plant a weigh-bridge or flow-meter captures milk volume delivered,

25

a delivery sample from the tanker provides percentage fat and protein, chemical and micro analysis data,

30 a laboratory sample provides data relating to pH, acidity, and anti-biotic content, and

- 2 -

silo control systems control flow and capture data for delivery to various production plant such as cheese-making or liquid milk production plant.

- 5 While there is a high level of automation at specific stages of production, heretofore provision of comprehensive production control and traceability from raw material source to retail outlet has been difficult and time-consuming to achieve.

The invention addresses this problem.

10

SUMMARY OF THE INVENTION

According to the invention, there is provided a traceability system comprising means for interfacing with food product supply chain stage devices to capture production data, and a traceability engine comprising means for storing production data in a manner whereby data from supply chain stages of a product is linked, characterised in that, the traceability engine comprises means for modelling the supply chain and for storing the production data in a structure of the model.

- 15 20 In one embodiment, the traceability engine comprises means for using the model to verify received production data before storing it.

In another embodiment, the traceability engine comprises an object representing each supply chain stage for which production data is generated, and methods linking the objects.

25

In a further embodiment, the model comprises means for using a method to link objects in any one selected direction.

- 3 -

In one embodiment, the traceability engine comprises means for writing a series of trace events to perform a trace, each of said trace events comprising a source object identifier and a target object identifier.

- 5 In another embodiment, the engine comprises means for using a relevant method to determine which source and target object identifiers are to be recorded as trace events.

- 10 In one embodiment, the traceability engine comprises means for maintaining a database of trace event records, the database structure being non-normalised.

- In another embodiment, the engine comprises means for generating a trace in response to a user request which defines an initial production stage object and a trace direction.

- 15 In a further embodiment, the engine comprises means for associating a particular method with the defined direction.

- 20 In one embodiment, the engine comprises means for associating a method to a selected direction in a two-dimensional map.

In another embodiment, the engine comprises means for generating a trace by retrieving and linking trace events.

- 25 In a further embodiment, the engine comprises means for retrieving and linking the trace events with recursive accesses to the trace events database.

In one embodiment, the system comprises means for automatically polling production control systems to retrieve production data, whereby modification of the

- 4 -

production control systems is not required for interfacing with the traceability system.

5 In another embodiment, the system further comprises an interface comprising means for pre-processing production data from continuous production devices before processing by the engine.

10 In a further embodiment, said interface comprises means for executing rules controlling recordal of multiple trace events for a single physical event.

In one embodiment, the interface comprises means for transmitting to the engine data representing a proportional value for a link between source and target records.

15 In another embodiment, the system comprises means for outputting a trace in a format similar to that of conventional operating system file/folder nested displays.

Detailed Description of the Invention

20 The invention will be more clearly understood from the following description of some embodiments thereof, given by way of example only with reference to the accompanying drawings in which:-

25 Fig. 1 is a high-level diagram illustrating a traceability system and its major interactions;

Fig. 2 is a more detailed diagram showing data capture.

Fig. 3 is a diagram illustrating the basic structure of a model of the system;

- 5 -

Fig. 4 is a diagram illustrating structure of a database controlled by the engine;

5 Fig. 5 is a screen shot of a list of objects, Fig. 6(a) is a screen shot displaying properties and descriptions of an object, and Fig. 6(b) is a screen shot showing attributes of an object property;

Fig. 7 is a screen shot showing linking of objects by methods, and Fig. 8 is a screen shot of method attributes;

10 Figs. 9(a) and 9(b) are screen shots showing trace enquiry functions;

Figs. 10 and 11 are displays of forward and reverse traces, and Fig. 12 is a diagram showing the two-dimensional nature of trace maps.

15 Description of the Embodiments

Referring to Fig. 1 a traceability system 1 performs food traceability and production control operations for a milk powder supply chain 2. The supply chain 2 comprises a
20 “discrete domain” in which the food materials are handled on a discrete basis, and a “continuous domain” in which there is continuous production to provide packed powder.

The traceability system 1 captures data and stage tags at each of the following stages
25 in the discrete domain:-

animal feed supply 3,

farm identification and production 4,

milk tanker identification and use 5,

- 6 -

milk load data 6,

factory intake bay identification and use 7, and

silo batch identification and use 8.

- 5 For each successive stage the system 1 updates a traceability events table so that it can provide a full history for the food at any particular stage. In addition, the system 1 assists production control by delivering a control signal to the intake bay stage to indicate which bay should be used. This is one example of a trace event.
- 10 The stages in the continuous domain in the factory are separation 9, skimming 10, evaporation/drying 11, and powder packing 12. These stages are controlled by a process control system with little input from the traceability system 1, other than quality alerts should the need arise. However, there is comprehensive data capture from the process control system 15. The traceability system 1 uses the captured data
- 15 to communicate with a laboratory information management system (LIMS) 25 and with an inventory system 20.

Referring now to Fig. 2 the manner of data capture for some of the discrete stages is shown in more detail. A farm computer 30 is used to access the system 1 to upload

20 data concerning the feed being used, and the farm itself such as farm identification and co-operative scheme data. A tanker automatic data capture (ADC) unit uploads tanker and load identifiers with QC update data from the LIMS system 25 and weighbridge and "swipe" card data is used by the traceability system 1 to download an intake bay selection to the tanker.

25

The system 1 operates independently of the data sources. This is because it comprises a polling interface which proactively pulls third party systems to capture data as it is generated. Also, it interfaces with internal or related applications without necessarily polling. The traceability engine therefore encapsulates trace

- 7 -

data. It requires no direct link to third party applications which create the production data. Also, such third party applications do not need to be modified to supply production data to the system 1.

- 5 Referring now to Figs. 3 to 10 the traceability system 1 is now described in more detail. The system 1 comprises an engine which models the supply chain using:

objects representing data records for supply chain stages, and
methods linking objects according to trace links.

- 10 In the context of any one method, the originating object is referred to as a "source" object and the destination one is referred to as a "target object".

The engine performs automatic verification of received data on the basis of checking that it matches allowable properties of the relevant object. The data received must
15 match the properties defined for the relevant objects. In addition, each incoming data property must conform to the object properties defined. For example the Finished Product Object is defined as:

Product : Type = Alpha : Length = 10

Batch : Type = Alpha : Length = 20

20 Pallet : Type = Alpha : Length = 10

If a 'FIN_PROD' trace event is invoked and the Product contains 15 characters then an error log will be created and the event will be rejected. In addition to checking the individual fields, the engine will check that this is a valid value for that object. So for
25 example, the Product/Batch/Pallet must exist as a record in the Inventory table.

The verification described above could be termed as 'passive' validation. However the structure of an events database allows for much more powerful 'active' validation. This is done by applying business rules such as the following example.

- 8 -

A trace method 'DISPATCH' is used to record a dispatch of finished goods. Suppose that as a strict rule, in order for the cheese to mature, the finished good must be left in the warehouse for at least one month. When the trace event is received, the
5 system will check that:

a value for the finished good exists in the events database, and that
this value was recorded more than one month ago.

As many business rules as required, specific to the business type can be added to the
10 validation process without affecting the generic functionality of the trace engine.

In more detail, referring to Fig. 3 there is an object class 30 for each of weighbridge, silo batch, makesheet column, IP (in-process batch) batch, and finished product supply chain stages. In the forward direction there is a method class 40 linking the
15 weighbridge and silo batch object classes and a single method class for each other object class as a source object class. The object classes and method classes are stored separately. Indeed, the method classes are also classes in the object-oriented paradigm as they are separate entities having attributes which in this case define both forward and reverse links.

20

The object classes have properties 50. For example, the makesheet (M/S) col. object class has Makesheet Header Type, M/S ID, and column no. properties. Each property has associated attributes.

25 In operation, when the engine receives data it checks it against the relevant object class properties for verification. If verification is positive it opens a record in a traceability events database 60, shown in Fig. 4. It writes an instance of an object class 30 to the record and writes data associated with each target object and writes this data to the same record together with data for the relevant method. In the

example of Fig. 4 there are two instances of the weighbridge object class, W/B01 and W/B02. In any particular supply chain there may be any relevant number of objects from a particular class. For example, a silo batch may fill three pallets and so a silo batch object is linked as a source object to three pallet objects.

5

The records of the database 60 are referred to as trace events. The structure of the database is not normalised, "normalised" in this specification meaning that an item is only stored once and is linked to other items by relationships between tables. By not normalising the structure, each trace event is complete and individually
10 addressable. Thus, all trace events can be processed in a universal manner. Also, the events can be linked to external systems such as the LIMs system in a direct manner. Non-normalisation also allows high-speed real time processing because there is no need to search through different tables for processing any one trace event.

15 Referring to Fig. 5, an attribute of some objects is an image for display purposes. The display of Fig. 5 shows for each object a name, sequence number, description and image properties. The sequence number is for display purposes only, the actual sequencing being determined by the methods.

20 Fig. 6(a) shows properties for the finished product ("FIN_PROD") object class. Object attributes are stored as data in a table. A verification routine uses the table name and an object string value to check that the record exists in the table. Fig. 6(b) shows attributes for the "Product" property of the FIN_PROD object.

25 Methods are illustrated in Fig. 7. Instantiation of a method involves recording the following data:-

source object identifier,

target object identifier, and

method description for the reverse or forward direction as appropriate.

- 10 -

As shown in Fig. 8, the definition of a method class includes source and target objects (in the forward direction sense), and a description for each of the forward and reverse directions.

5

Once trace events have been recorded, the engine generates a trace in response to a request, as follows:-

10

it allows the user to input an object for the starting-point production stage, such as "SUP-VOLUME as shown in Fig. 9(a), and a particular instance using a look-up screen as shown in Fig. 9(b),

15

uses the available methods to generate prompts for the user to choose a particular method based on the production stage and choice of one available direction from that stage, also as shown in Fig. 9(a) and described in more detail below,

20

retrieves the object associated with that stage and treats it as a source object for a forward trace, or as a target object if it is a reverse trace,

25

retrieves the linked target (or source) object(s), and

retrieves the trace events for the objects, the trace events having been previously recorded during production in real time, and displays the trace event data.

A forward trace view is shown in Fig. 10, and a reverse trace view is shown in Fig. 11. These views can be readily generated by simply directly outputting the contents of the string of trace events for the supply chain in question. There is no need to

- 11 -

search through relational tables and build a trace. It simply exists in non-normalised form, ready for direct outputting.

5 This default trace direction is forward but the user can change this to a reverse direction in making the trace request. The selection is not simply forward or reverse; A trace chain is not necessarily one dimensional – it could well be two dimensional. For example a Silo could receive the main raw material – ‘Wholemilk’ but it could also have some ingredients mixed in – for example ‘Salt’. If the user requests a trace from the Silo then they have three choices:

- 10 Forward to the processing stage
Reverse to the receipt of Wholemilk
Reverse to the receipt of Good In / Ingredients.

This is illustrated diagrammatically in Fig. 12.

- 15 The user provides a date range so that the next lookup on trace events is limited to this date range (From/To)

The user supplies a range of trace events to run the trace engine on. Providing a range allows them to for example select a range of pallets to trace if they were not
20 sure which pallet the problem originated from.

The description above explains at a high level how the engine generates a trace in response to a trace request. In more detail, the following is a sample extract from the events database 60, in which each row is an event record. The second and fourth
25 columns directly include sufficient data for the trace. However, should additional data be required for other purposes the values in the second and fourth columns are keys to other tables within the system.

Table 1

Source		Target	
Object	Value	Object	Value
Weighbridge Docket	90001	Silo	SMP~20045
Weighbridge Docket	90002	Silo	SMP~20045
Weighbridge Docket	90003	Silo	SMP~20045
Silo	SMP~20045	In Process Batch	IP483
Silo	SMP~20046	In Process Batch	IP483
In Process Batch	IP483	Finished Product	CHEDDAR~T205~C1230001
In Process Batch	IP483	Finished Product	CHEDDAR~T205~C1230002
In Process Batch	IP483	Finished Product	CHEDDAR~T205~C1230003
In Process Batch	IP483	Finished Product	CHEDDAR~T205~C1230004
In Process Batch	IP483	Finished Product	CHEDDAR~T205~C1230005
In Process Batch	IP483	Finished Product	CHEDDAR~T205~C1230006
Finished Product	CHEDDAR~T205~C1230001	Dispatch	321~ CHEDDAR~T205~C1230001
Finished Product	CHEDDAR~T205~C1230002	Dispatch	321~ CHEDDAR~T205~C1230002

- Take a forward trace from Weighbridge Docket 90001. The value on the left hand side is read and the corresponding value on the right hand side is stored and used as
- 5 a source value for the next step in the trace chain. Thus , this will retrieve the following trace events:

Source		Target	
Object	Value	Object	Value
Weighbridge Docket	90001	Silo	SMP~20045
Silo	SMP~20045	In Process Batch	IP483
In Process Batch	IP483	Finished Product	CHEDDAR~T205~C1230001
In Process Batch	IP483	Finished Product	CHEDDAR~T205~C1230002
In Process Batch	IP483	Finished Product	CHEDDAR~T205~C1230003

- 13 -

In Process Batch	IP483	Finished Product	CHEDDAR~T205~C1230004
In Process Batch	IP483	Finished Product	CHEDDAR~T205~C1230005
In Process Batch	IP483	Finished Product	CHEDDAR~T205~C1230006
Finished Product	CHEDDAR~T205~C1230001	Dispatch	321~ CHEDDAR~T205~C1230001
Finished Product	CHEDDAR~T205~C1230002	Dispatch	321~ CHEDDAR~T205~C1230002

Now take a reverse trace starting from a Dispatched pallet. The logic is exactly the same but reversed, i.e. the value on the LHS is read and the corresponding value on the RHS is stored and used as a target value for the previous step in the trace chain.

- 5 Thus, this will produce the following trace events:

Target		Source	
Object	Value	Object	Value
Dispatch	321~ CHEDDAR~T205~C1230001	Finished Product	CHEDDAR~T205~C1230001
Finished Product	CHEDDAR~T205~C1230006	In Process Batch	IP483
In Process Batch	IP483	Silo	SMP~20045
In Process Batch	IP483	Silo	SMP~20046
Silo	SMP~20045	Weighbridge Docket	90001
Silo	SMP~20045	Weighbridge Docket	90002
Silo	SMP~20045	Weighbridge Docket	90003

- Referring again to Figs. 10 and 11, the trace views are presented in the conventional operating system folder/file nested display manner. This is very easy for the user to understand and allows for collapsing/expanding a trace chain over multiple production stages.

- It will be noted that the screens of Figs. 10 and 11 have "Forward" and "Reverse" buttons. These allow the user to recursively request a trace path based on values outputted for a previous trace request. This is particularly useful for a business

- 14 -

scenario where the following steps must be taken, (a) contamination of end product notified, (b) reverse trace request up the supply chain, and (c) forward trace to see what other end products must be recalled.

- 5 The above describes traceability for the discrete domain, in which production stages are clearly defined with definite boundaries. For the continuous domain the stages are often not so clearly defined. For example one pallet of powder may have powder derived from multiple silos because of residual powder and milk in pipework. The engine includes an interface layer of rules which initially process the incoming
- 10 production data from the continuous domain. These rules control and direct recordal of multiple trace events for some single physical events in the continuous domain. For example, production data for ejection of a powder pallet results in the rules linking the pallet eject object with multiple silo batch objects according to user-defined configuration settings. One such setting is a percentage residual powder
- 15 quantity based on time for a particular silo.

In one example, the user may define the following 'bands' of percentage composition against the lag time from whole milk flowing from a silo to ending up as finished powder.

20

No	Description	Composition Percentage	Lag Time
1	Main Flow	80%	20
2	Silo Open / Shut Overlap	15%	30
3	Residue 1	4%	40
4	Residue 2	1%	720

The above tables should be read as follows:

Taken one Pallet ejected , the user expects that:

25

- 15 -

80% of the powder contained within that pallet can be derived from a silo opened at least 20 minutes ago (because that's how long the main flow takes),

5 a further 15% of the powder contained within that pallet can be derived from a silo opened at least 30 minutes ago (because of the Open/Shut overlap between the current and previous silo),

10 a further 4% of the powder contained within that pallet can be derived from a silo opened at least 40 minutes ago (because of a large amount of residue still in the system),

a further 1% of the powder contained within that pallet can be derived from a silo opened at least 12 hours ago (because of a small amount of residue still in the system)

15

From the user defined parameters above, the interface layer will for one physical trace event – in this case the creation of a pallet of bags of powder – create multiple trace events but with varying proportions.

20 This interface allows the core traceability engine functionality to operate in a common universal manner irrespective of whether the data originates in the continuous or discrete domains.

25 It will be appreciated that the invention provides for comprehensive capture of full supply chain data with excellent verification to ensure data integrity. Trace events are dynamically created in real time according to the simple object and method classes.

- 16 -

There is excellent versatility because of availability of multiple forward and multiple reverse directions, and because the system processes data from the discrete and continuous domains in a universal manner.

- 5 The invention is not limited to the embodiments described but may be varied in construction and detail.

Claims

1. A traceability system (1) comprising means for interfacing with food product supply chain stage devices (3-12) to capture production data, and a traceability engine comprising means for storing production data in a manner whereby data from supply chain stages of a product is linked, characterised in that, the traceability engine comprises means for modelling the supply chain (30, 40, 50) and for storing the production data in a structure of the model.
2. A traceability system as claimed in claim 1, wherein the traceability engine comprises means for using the model (30, 40, 50) to verify received production data before storing it.
3. A traceability system as claimed in claims 1 or 2, wherein the traceability engine comprises an object (30) representing each supply chain stage for which production data is generated, and methods (40) linking the objects.
4. A traceability system as claimed in claim 3, wherein the model (30, 40, 50) comprises means for using a method to link objects in any one selected direction.
5. A traceability system as claimed in claims 3 or 4, wherein the traceability engine comprises means for writing a series of trace events to perform a trace, each of said trace events comprising a source object identifier and a target object identifier.
6. A traceability system as claimed in claim 5, wherein the engine comprises means for using a relevant method to determine which source and target object identifiers are to be recorded as trace events.

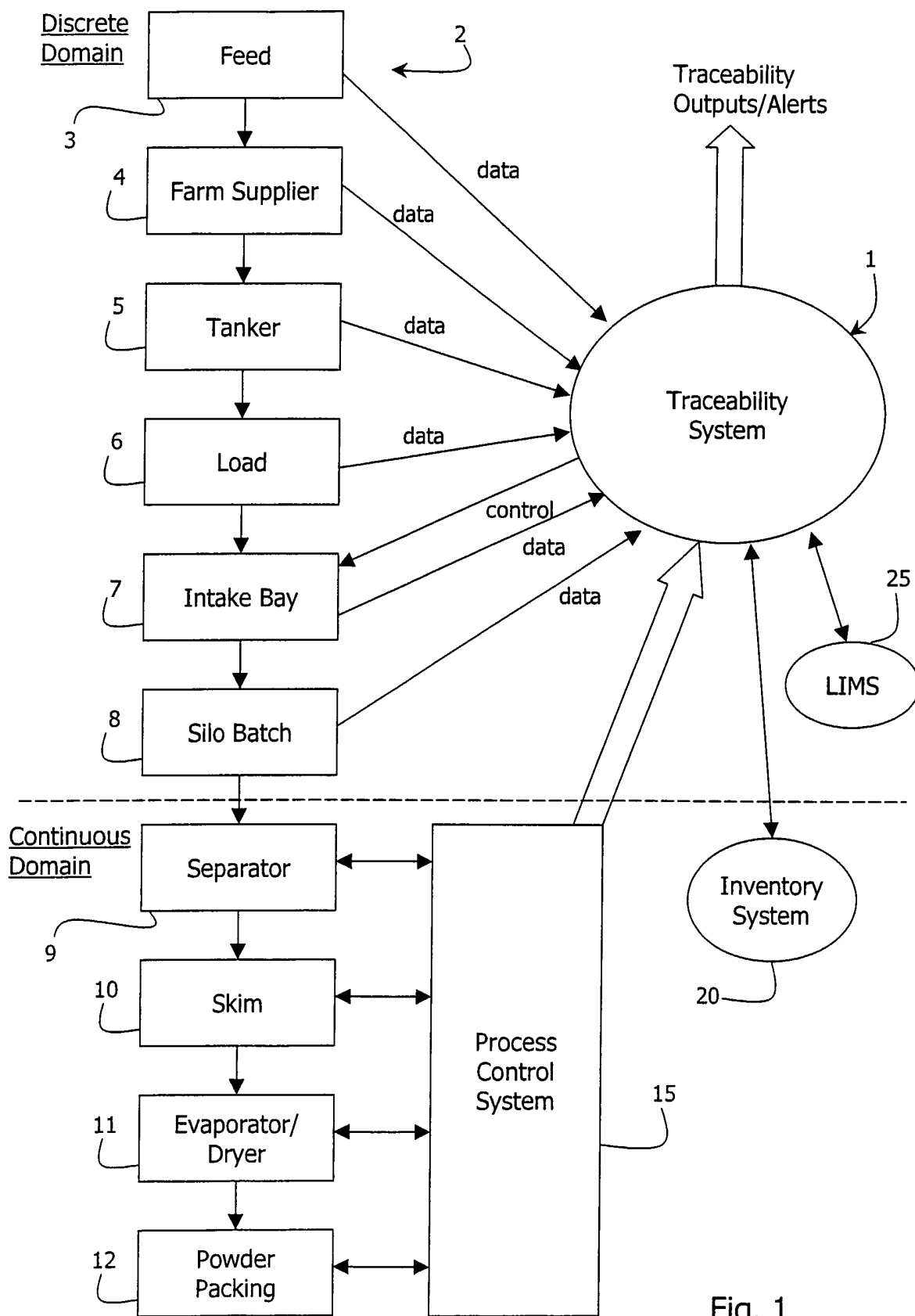
- 18 -

7. A traceability system as claimed in claims 5 or 6, wherein the traceability engine comprises means for maintaining a database (60) of trace event records, the database structure being non-normalised.
- 5 8. A traceability system as claimed in any preceding claim, wherein the engine comprises means for generating a trace in response to a user request which defines an initial production stage object and a trace direction.
9. A traceability system as claimed in claim 8, wherein the engine comprises
10 means for associating a particular method with the defined direction.
10. A traceability system as claimed in claim 9, wherein the engine comprises means for associating a method to a selected direction in a two-dimensional map.
15
11. A traceability system as claimed in any of claims 8 to 10, wherein the engine comprises means for generating a trace by retrieving and linking trace events.
12. A traceability system as claimed in claim 11, wherein the engine comprises
20 means for retrieving and linking the trace events with recursive accesses to the trace events database.
13. A traceability system as claimed in any preceding claim, wherein the system comprises means for automatically polling production control systems to
25 retrieve production data, whereby modification of the production control systems is not required for interfacing with the traceability system.
14. A traceability system as claimed in any preceding claim, wherein the system further comprises an interface comprising means for pre-processing

production data from continuous production devices before processing by the engine.

- 5 15. A traceability system as claimed in claim 14, wherein said interface comprises means for executing rules controlling recordal of multiple trace events for a single physical event.
- 10 16. A traceability system as claimed in claim 15, wherein the interface comprises means for transmitting to the engine data representing a proportional value for a link between source and target records.
- 15 17. A traceability system as claimed in any preceding claim, wherein the system comprises means for outputting a trace in a format similar to that of conventional operating system file/folder nested displays.
18. A computer program product comprising software code for completing a traceability system as claimed in any preceding claim, when executing on a digital computer.

1/10



2/10

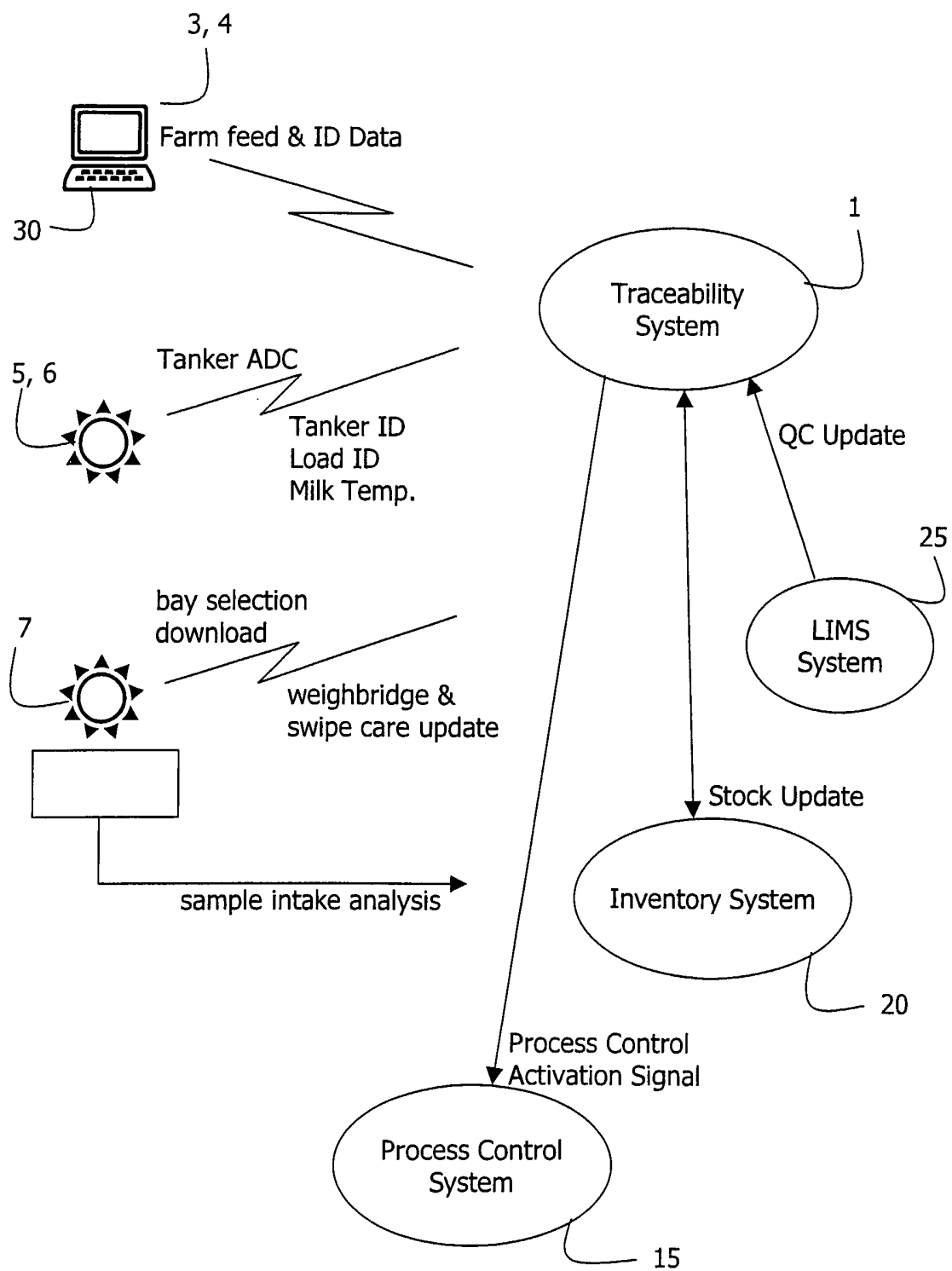


Fig. 2

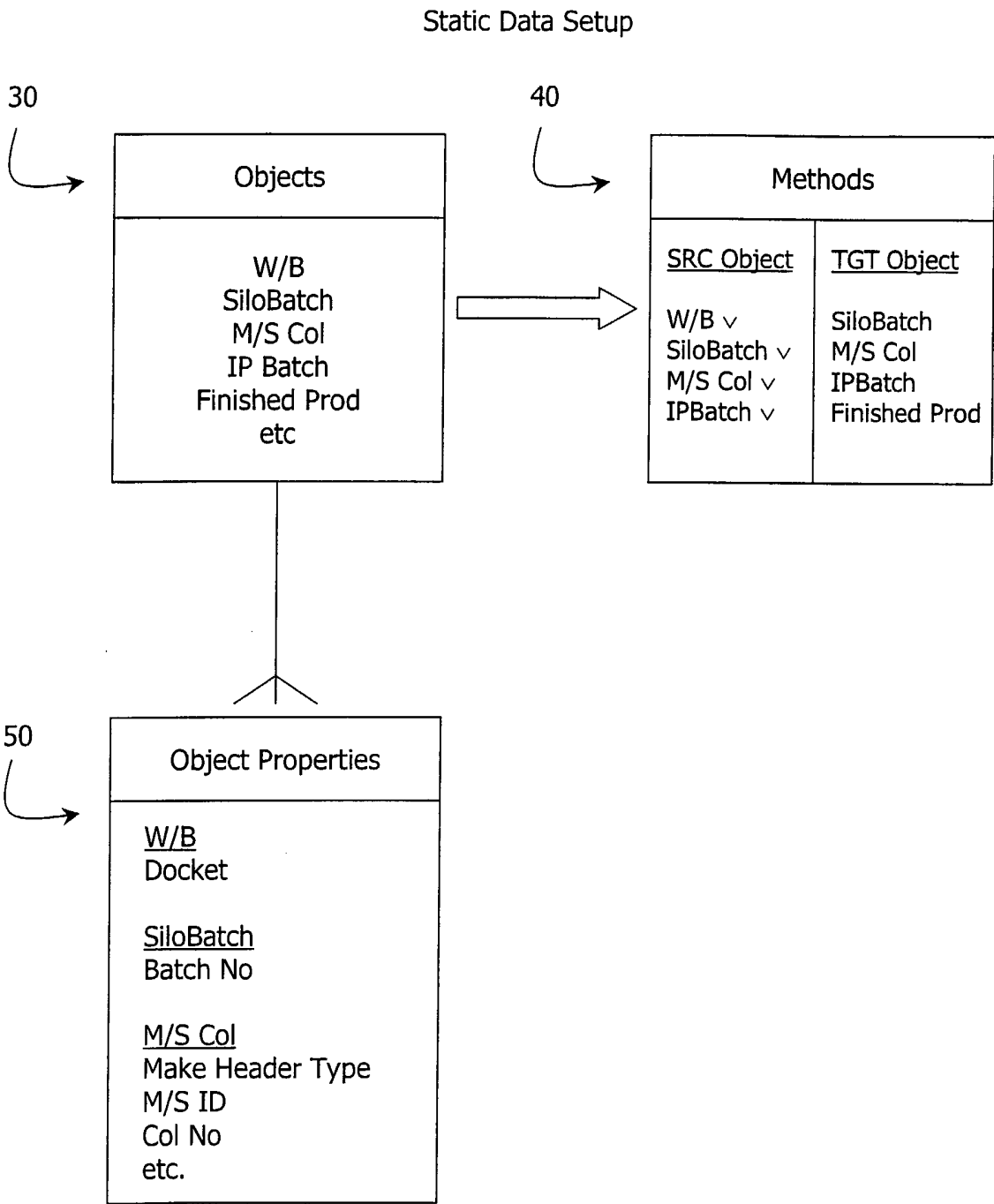


Fig. 3

4/10

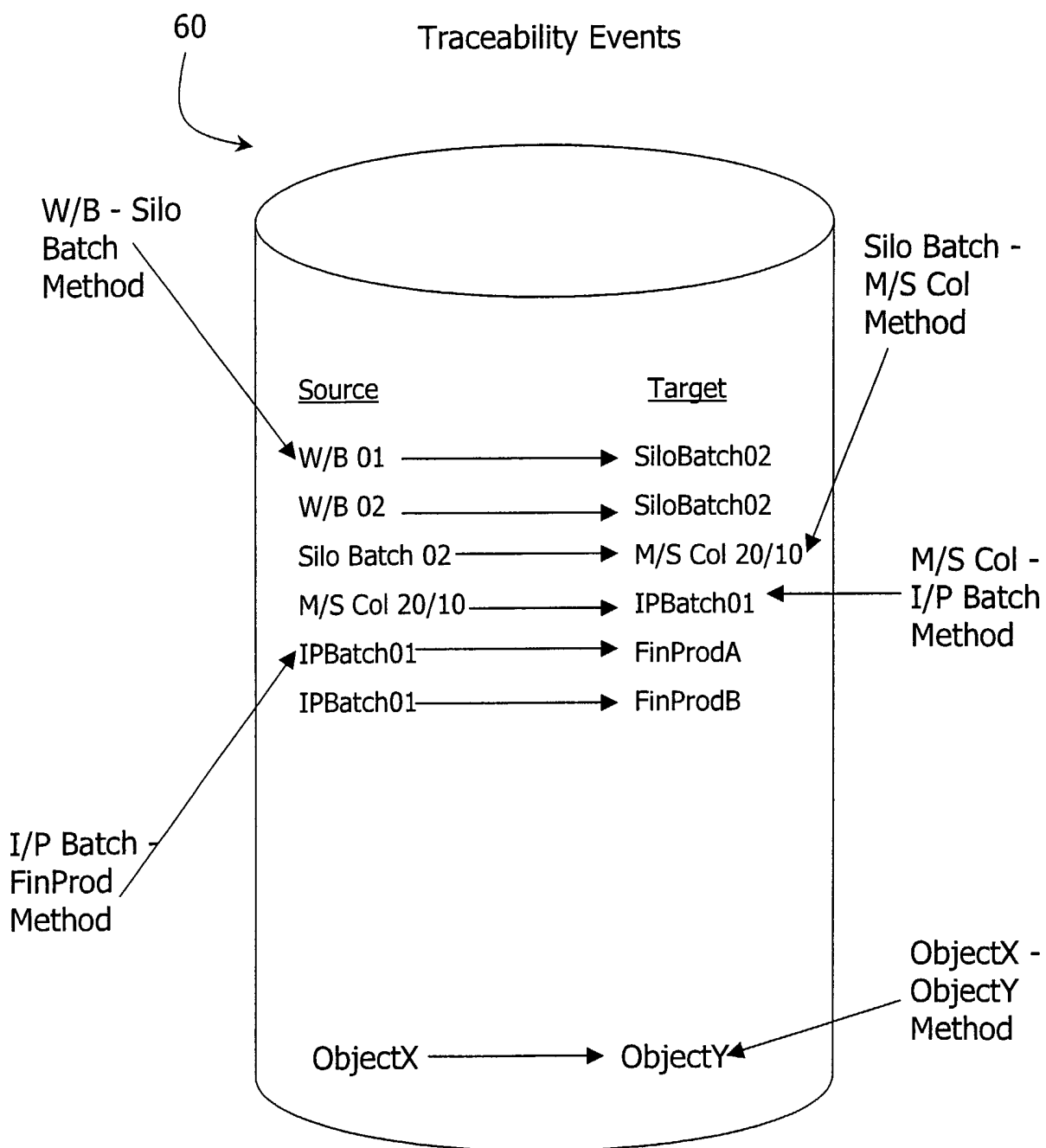


Fig. 4

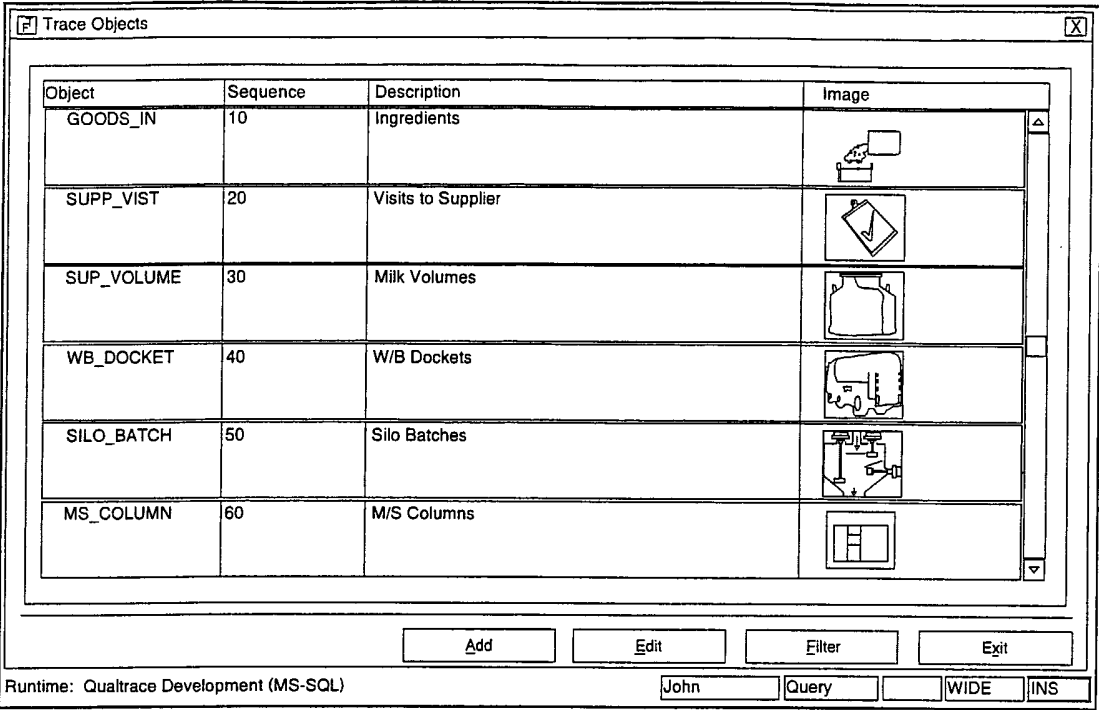


Fig. 5

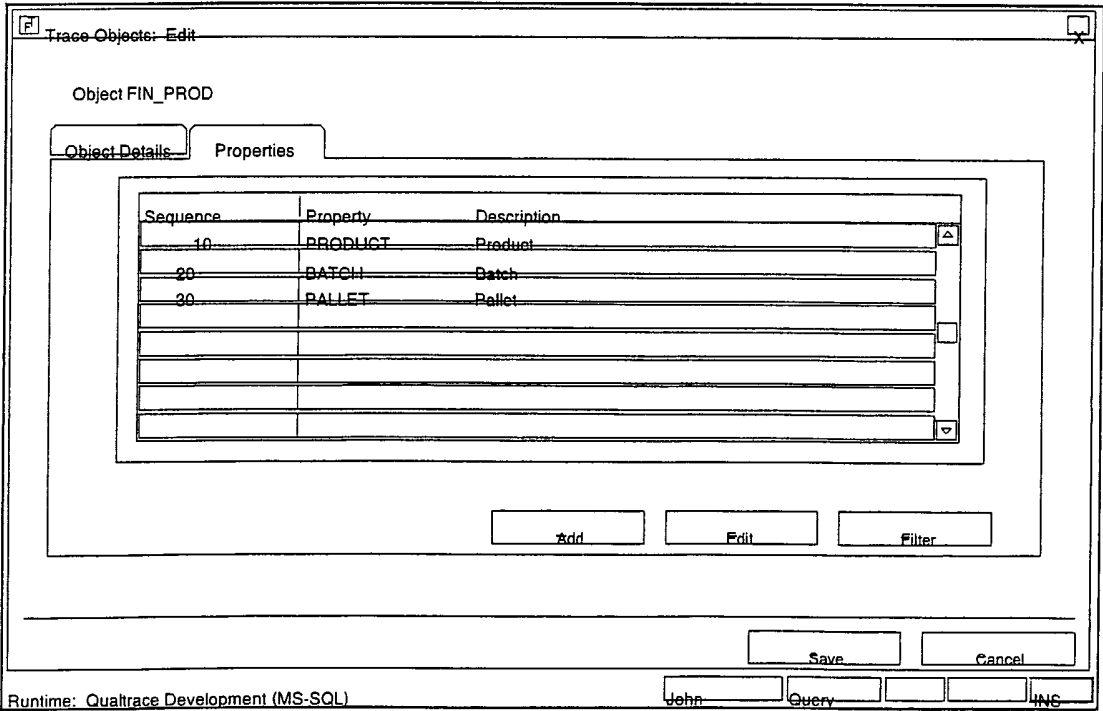


Fig. 6 (a)

6/10

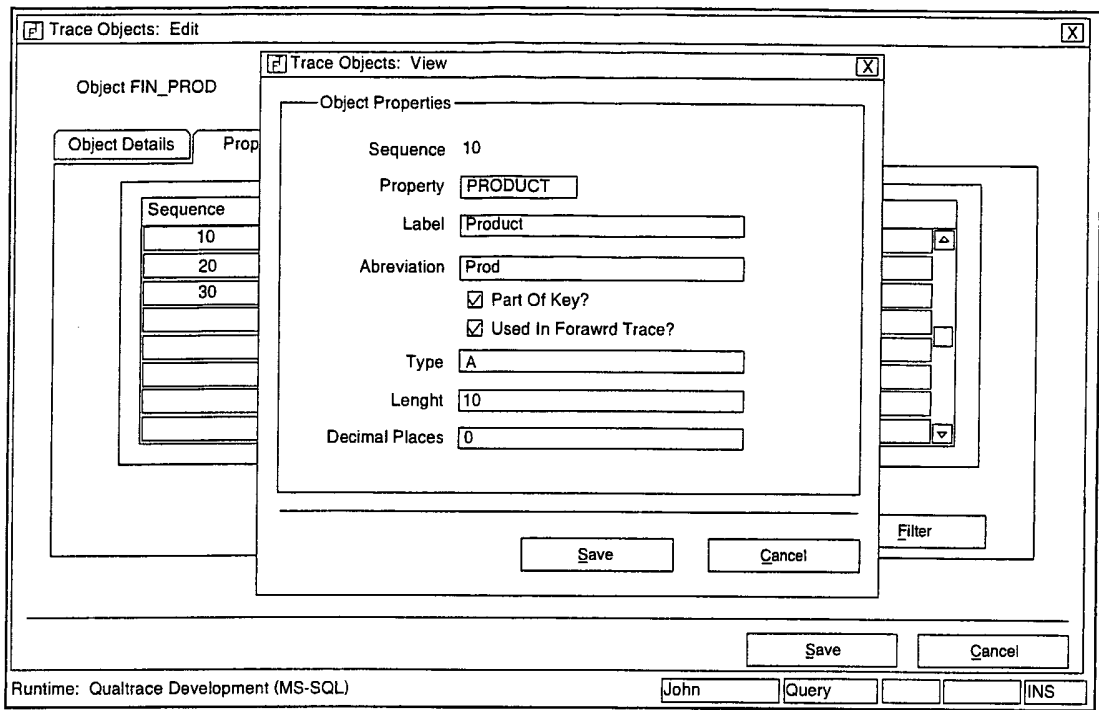


Fig. 6(b)

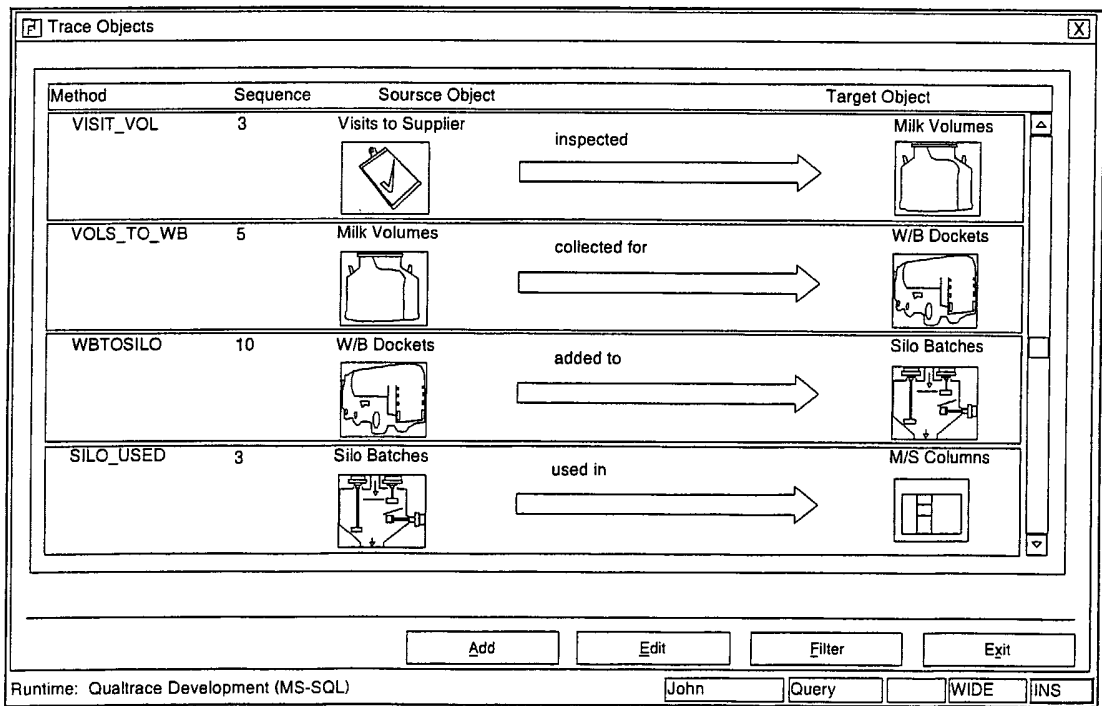


Fig. 7

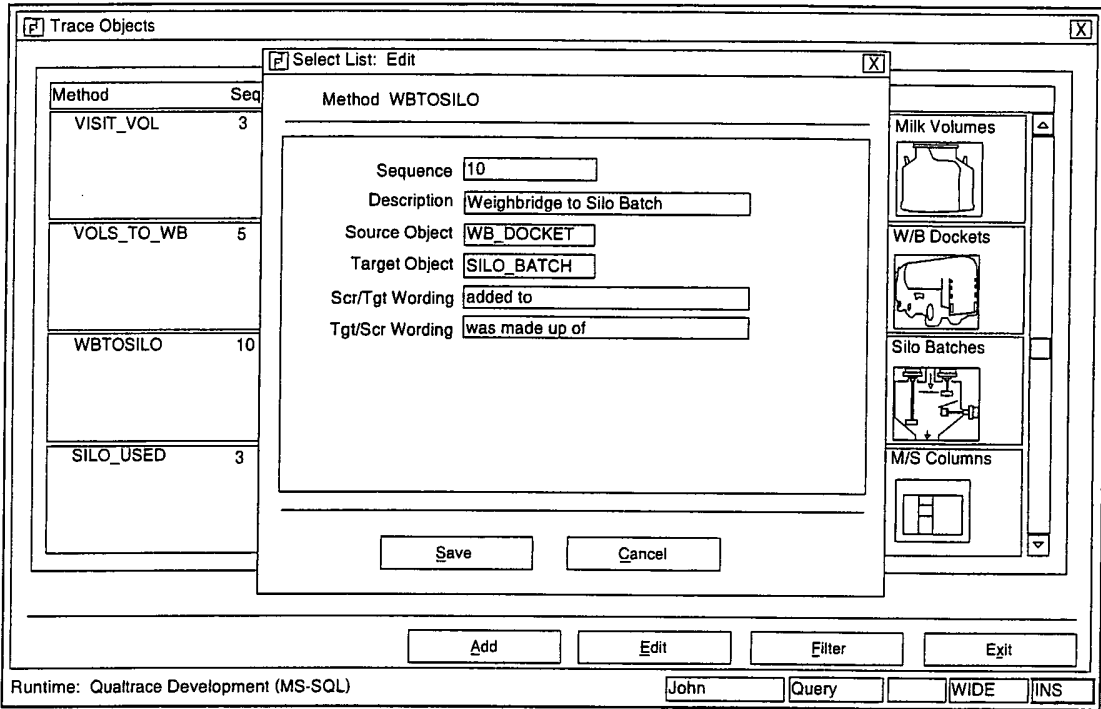


Fig. 8

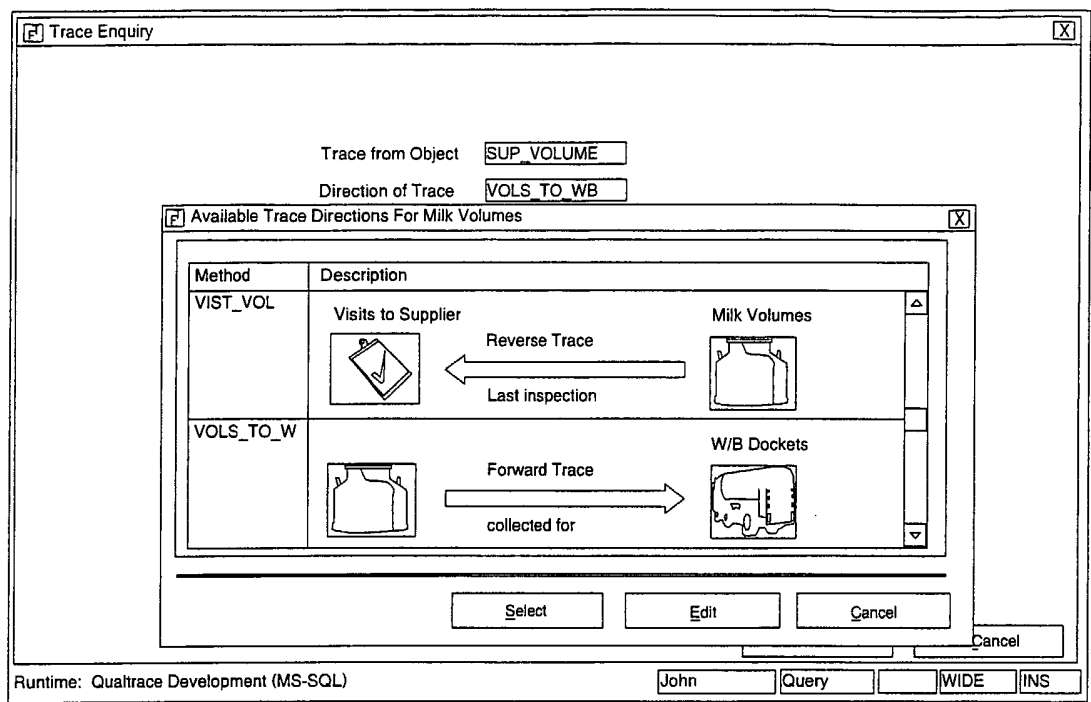


Fig. 9 (a)

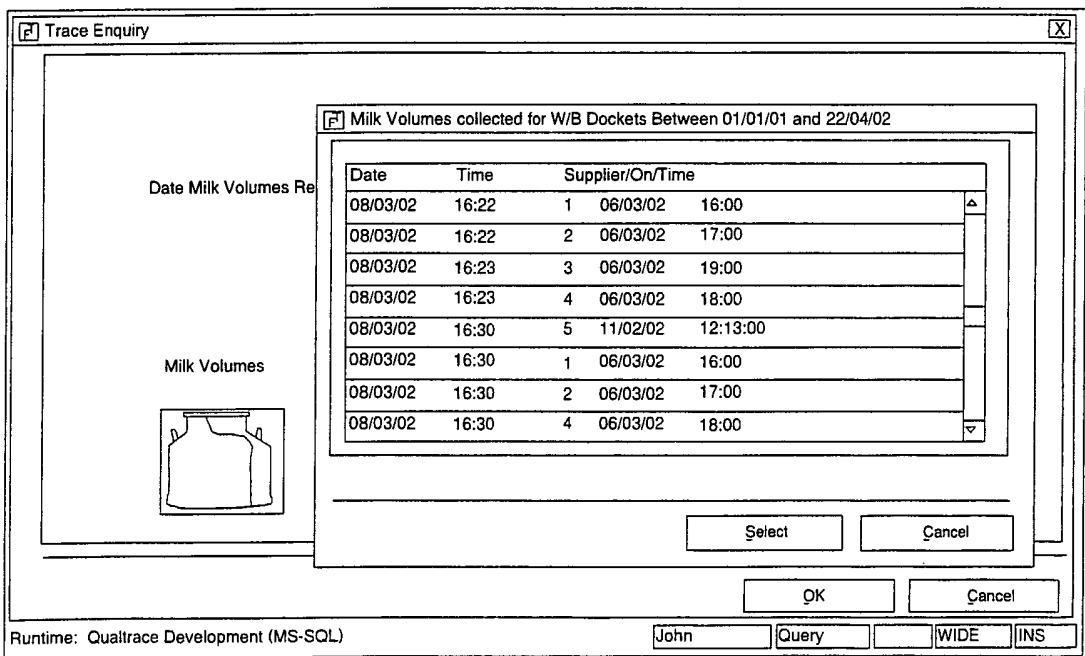


Fig. 9 (b)

9/10

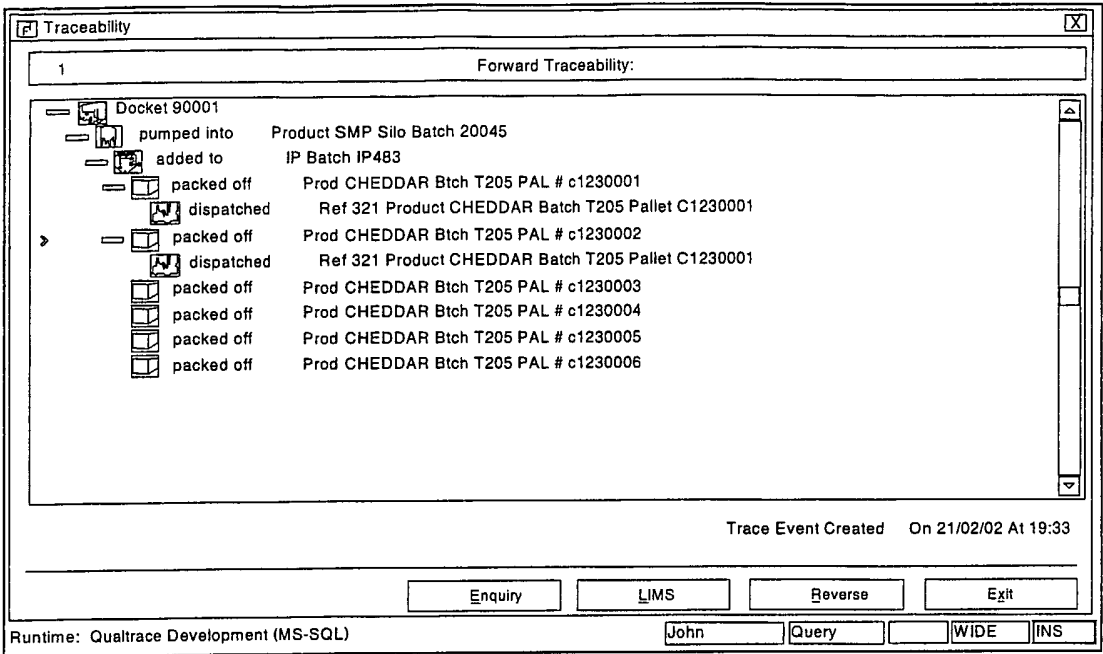


Fig. 10

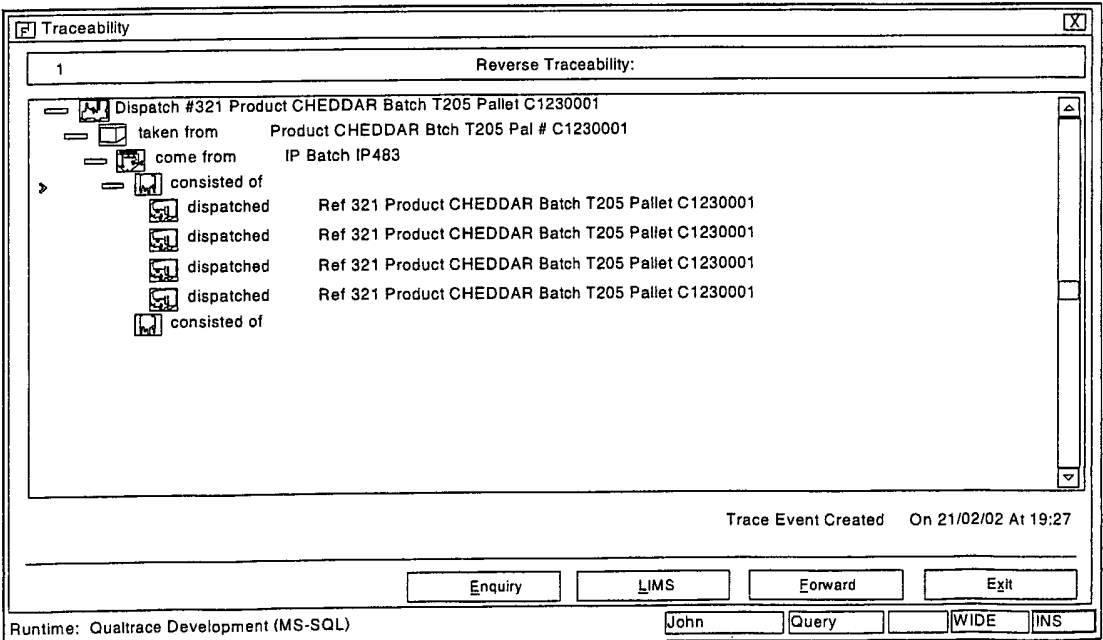


Fig. 11

10/10

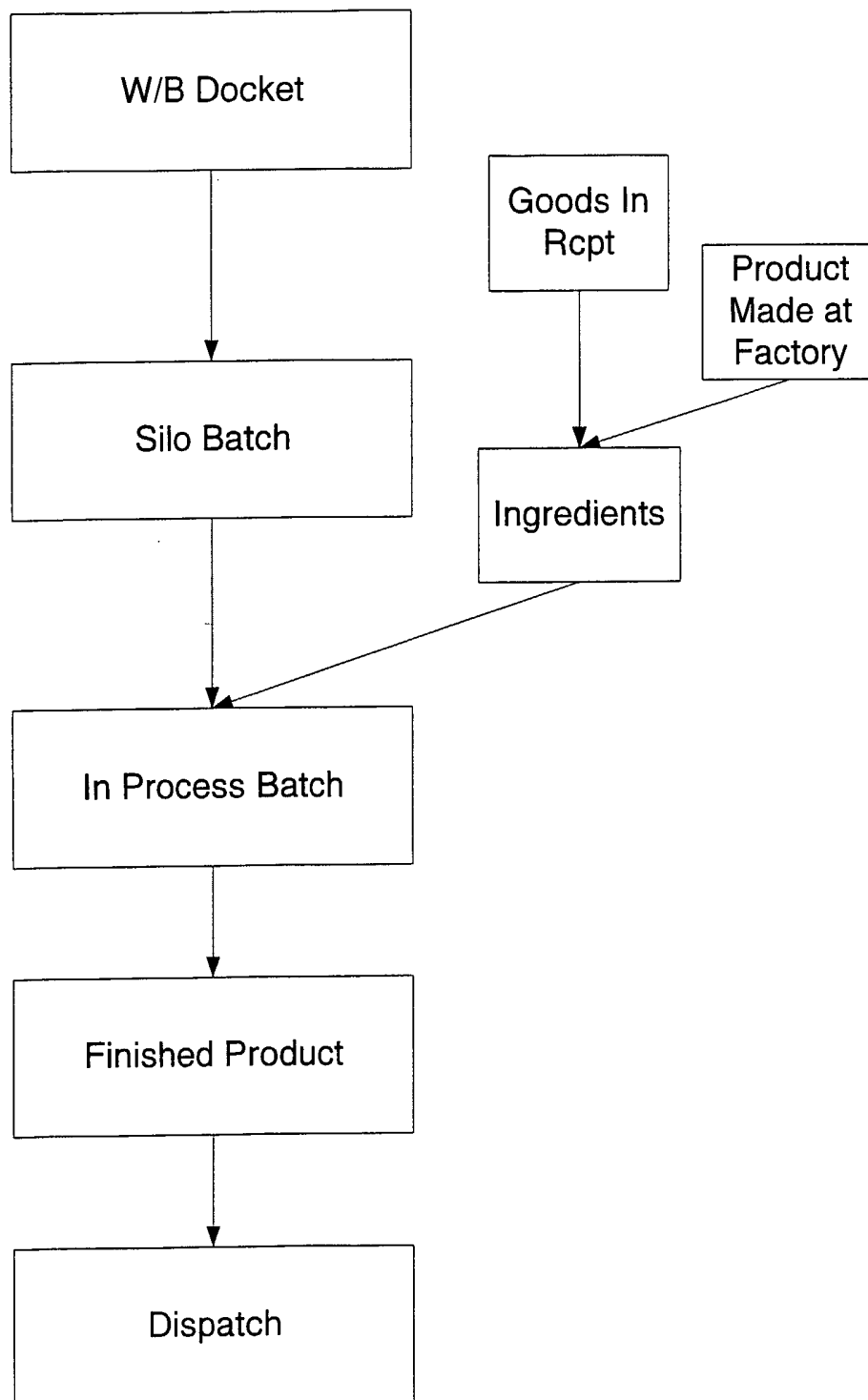


Fig. 12