# WEB SERVICE-BASED TRACEABILITY IN FOOD SUPPLY CHAINS

Andréia Akemi Kondo, Claudia Bauzer Medeiros,
Edmundo Roberto Mauro Madeira, Evandro Bacarin
*Institute of Computing, University of Campinas, Campinas, São Paulo, Brazil*

Keywords: Traceability, Supply Chains, Workflows, Web Services.

Abstract: Supply Chains present many research challenges in Computing, such as the modeling of their processes, communication problems between their components, logistics and processes management. This paper presents a supply chain traceability model that relies on a Web service-based architecture to ensure interoperability. Geared towards assisting quality control in the agricultural domain, the model allows to trace products, processes and services inside a chain. It has been validated for real life case studies and the Web service implementation is under way.

## 1 INTRODUCTION

A supply chain is a network of retailers, distributors, transporters, storage facilities and suppliers that participate in the sale, delivery and production of a particular product (Kumar, 2001; Min and Zhou, 2002). It is composed of distributed, heterogeneous and autonomous elements, whose relationships are dynamic.

Figure 1 presents a basic example of the milk supply chain. Starting from a product (input) – (*Milk 1*) – it produces another product – (*Cheese*), going through transformations, storage and transportation steps. The figure shows units of Production (*Milk Producer 1 and Dairy 1*), Transportation (*Transp1*, *Transp2* and *Transp3*) and Storage (*Warehouse 1 and Supermarket 1*). Any supply chain can be represented by a directed graph which has at least one source – e.g., (*Milk Producer 1*) and one sink (*Supermarket1*). The back arrow represents a return flow, where some outputs of a chain can be used as inputs of the same or other chains – e.g., milk overflowing from vats can be used to enrich food for the animals at the farm.

Traceability in food supply chains is gaining importance everywhere all over - be it to feed humans, plants or animals. The food industry is investing massively in issues involving health – e.g., safety, storage, manipulation and composition of products.

In particular, meat supply chains have attracted attention from all over the world, with the appearance of the Creutzfeldt-Jacob Disease (also known as the *mad cow* disease). New devices have been devised to tag animals, and special hardware and software systems keep track of specific events in the animal's life. The emphasis, however, is on constructing such devices. What is needed is a generic computational infrastructure to support traceability in a general way.

In Computer Science, supply chain traceability management involves many challenges, such as capturing, identifying and storing relevant events, managing associated constraints, analyzing the interaction of actors in a chain, monitoring negotiation protocols, and many others. Though some of these computing issues are studied in the context of supply chains, food supply chain traceability is still a relatively new field.

This paper contributes to solving this problem, by presenting a traceability model for agriculture supply chains. This model is based on two main concepts: (1) using distributed repositories to store sets of records concerning specific events along a given food supply chain; and (2) defining Web services to access these repositories, thereby transforming the problem of traceability derivation into that of performing a set of service invocations.

## 2 RELATED WORK

### 2.1 Supply Chains and Base Model

An agricultural supply chain is composed of all the activities that occur from the production to the fi-



Akemi Kondo A., Bauzer Medeiros C., Roberto Mauro Madeira E. and Bacarin E. (2007).
WEB SERVICE-BASED TRACEABILITY IN FOOD SUPPLY CHAINS.
In *Proceedings of the Third International Conference on Web Information Systems and Technologies - Internet Technology*, pages 171-177
DOI: 10.5220/0001270701710177

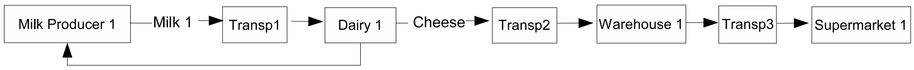


Figure 1: Basic example: the Milk Supply Chain.

nal consumer. The flow inside an agricultural supply chain is subject to several controls, most of which can affect human health. Such controls must be enacted at all phases within the chain, complicating its monitoring.

Our work is based on a general model (Bacarin et al., 2004) composed of the basic components: Actors, Production, Storage and Transportation.

Chain dynamics are modeled by Regulations, Contracts, Coordination Plans and Summaries. Regulations are restrictions imposed at several phases of the chain. Contracts are agreements between trading partners. Coordination Plans are directives that describe chain execution, coordinating the interactions between its components. Summaries are elements introduced for traceability and auditability. This paper is concerned with traceability issues.

## 2.2 Traceability

Traceability is a concept that involves several types of activities and refers to the ability to describe and to follow the life of a conceptual or physical element, preserving its identity and its origins. In database systems, traceability is associated with execution of transaction logs. In software engineering, it is related to all phases of software development from the requirements to the final product. It is also present in fault tolerance studies or in system auditing.

Most scientific papers on agricultural traceability ignore implementation aspects. They are concerned with logistics (Thomas and Griffin, 1996) and strategies of chain execution (Guiffrida and Nagi, 2006) or certification (Stock, 2004). These papers center on the business at hand, and computing issues are not treated at depth.

More recently, work has appeared concerning a computer science perspective – e.g., (Bello et al., 2004) or (Cimino et al., 2005). The first proposes a distributed collaborative information system that uses XML for representing data and Web Services to interface distinct suppliers which communicate through a HTTP protocol. The second describes a traceability system that supports exchanging documents between several units of the supply chain. It uses ebXML (Electronic Business using eXtensible Markup Language), a set of specifications that enable companies to conduct business over the Internet.

According to (Opara, 2003), there are seven important elements in agricultural traceability: product, process, genetic constitution, inputs, disease, pests and measurement traceability. Many of these factors are associated with geographical coordinates. As a consequence, additional information can be derived - e. g., associated with cultural aspects, which may influence food preparation and preservation. An example of the use of coordinates is the GeoTraceAgri (Debord et al., 2001) european project, where geographic coordinates are the prime element for traceability studies. As will be seen, our model also relies on this information.

## 2.3 Workflows and Web Services

In our work, processes within a chain are modelled as workflows, in which an activity can encapsulate other activities. A workflow represents the automation of procedures where documents, information or tasks are passed between participants for executing a certain action according to a defined set of rules (Hollingsworth, 1995).

Figure 1, the basic supply chain, can be seen as a workflow where each box denotes a *Production* element (e.g, *Dairy*), a *Transportation* element or a *Storage* element. Each box comprises a wide range of activities that occurs within the supply chain. The *Dairy* can furthermore encapsulate another chain (workflow).

In our architecture, each workflow activity may be considered as invoking some sort of Web service. Web Services are self-describing and modular business applications that provide business logic as services over the Internet through standards-based interfaces and Internet protocols (e.g. HTTP) with the purpose of finding, subscribing and invoking those services (Nagappan et al., 2003; Alonso et al., 2004). These standards include XML (Extensible Markup Language), SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language) and UDDI (Universal Description, Discovery and Integration). Web Services facilitate the communication between distinct applications and platforms. As will be seen, our traceability model is also enacted via service invocation and composition.

# 3 THE TRACEABILITY MODEL

## 3.1 Model Overview

Our traceability architecture is concerned with recording relevant events that occur within an agricultural supply chain. Those events are incrementally registered in summaries (Section 3.2) that are stored in interlinked repositories (Section 3.3).

Let us introduce our three main summaries (Product, Process and Service) via our running example. Consider again Figure 1 and the cheese making process. This process demands a certain amount of milk, coagulating agent and salt. At the Dairy, those ingredients are mixed and undergo the process that produces cheese. Each ingredient has its own Product Summary. For instance, the milk summary stores information that allows discovering among other things where and when the milk was produced, if any milk cow received certain undesirable substances (a drug, a vaccine). The production process itself has several stages, such as: curd formation, cutting, cooking, salting, pressing and curing. Although this process applies to most kinds of cheese, there are variations and each process occurs under distinct condition. The information about each instance of the cheese making process (ingredients and specific conditions) is stored in a Process Summary.

The cheese making process produces two distinct products: cheese and whey. These products may be used by other production processes. Thus, it is necessary to create a specific Product Summary for each product. Finally, the cheese may be transported to a grocer and the whey to an animal food industry, where they may be stored. Thus, there is a Service Summary associated to each transportation and storage step.

## 3.2 Summaries

A Summary is a sequence of records that, similar to a database log, describes events of a supply chain for traceability purposes. There are 3 kinds of summaries:

- Product: it records the sequence of events that happen along the life of a product, from its creation to its final consumption.
- Process: it records the details and the phases of a production process. A process is any activity that transforms a set of inputs into one or more different products.
- Service: it records information about a service. A service is any activity that may influence a product, without transforming it, e.g. transportation or storage.

### 3.2.1 Product Summary

Figure 2 presents a general view of a Product Summary record, illustrating its links to other data sources (repositories or other summary records).

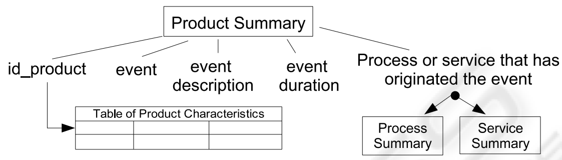


Figure 2: Product Summary Record: General View.

The field "id_product" uniquely identifies the product instance. The "events" field refers to phases of the supply chain, such as creation, storage, packing, mixture, consumption etc. Events are described by "Event description". The field "process or service that has originated the event" is a pointer to a record in the Process or Service Summary that describes the process or service which originated the event.

Records in the Product Summary refer to specific instances of a product. This summary is complemented by a table (Table 1) that defines the properties of an instance of the product. Each property is defined as a record composed of three fields, similar to an RDF description: "id", "property name" and "value". For example, quantity is a property of the product instance identified as *CheeseProd* (Table 1). The value of each property is specified by the field "value" (e.g. the value *circular* for property *shape*). This table allows managing products with distinct properties – here milk has three properties and cheese has four.

Table 1: Table of Product Characteristics.

| Table of Product Characteristics | | |
|---|---|---|
| **id** | **Property name** | **Value** |
| MilkProd | productsrep | MilkA |
| MilkProd | quantity | 10000 |
| MilkProd | validity date | 10/06/2006 |
| CheeseProd | productsrep | Minas |
| CheeseProd | quantity | 2000 |
| CheeseProd | validity date | 12/02/2006 |
| CheeseProd | shape | Circular |

### 3.2.2 Process Summary

Figure 3 presents a general view of a Process Summary record. Similar to what occurs with product instances, records in Process Summaries are instances of more general process descriptions stored in the Process Repository (see Section 3.3).

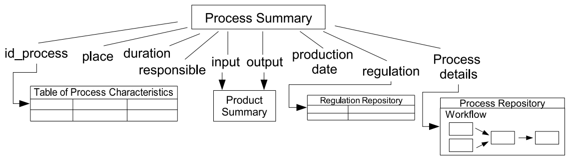


Figure 3: Process Summary Record: General View.

The "id_process" field identifies a process instance. Every time a process is executed, a new instance is generated. The "place" field indicates the geographical coordinates of the place where the process is executed; the "duration" of the process is a time interval characterized by two timestamps: process begin and process end; "responsible" indicates the entity (human, software or machine) in charge of the process; "production date" is the date in which the process created a product. The "input" field indicates the raw materials that are required for producing that product. The field "output" contains one or more identifiers of the products that are produced by the process; "regulation" contains references to the norms or conditions that are followed by the process; "process detail" points to a workflow, stored in the Process Repository, which presents details of the process (see Section 3.3). If a workflow encapsulates other more detailed workflows, additional records will be stored in the Process Summary, refining the process activities. This feature allows to record process execution in several granularity levels.

The Process Summary is complemented by a table that defines the characteristics of an instance of the process. This table is similar to the product characteristic table (Table 1).

### 3.2.3 Service Summary

Figure 4 presents a general view of a Service Summary record. A service can be classified as Transportation or Storage.

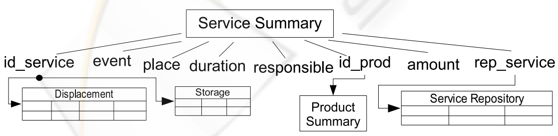


Figure 4: Service Summary Record: General View.

A Service Summary record is identified by a key composed of four fields: "id_service", " id_product", "place" and "event". Each record is related to an event that happens while the service is carried out – e.g., the loading or unloading of products (during transportation or storage). The field "id_service" indicates that a record is related to a specific service. "Amount" indicates the loaded or unloaded quantity; "id_prod" identifies the product that is the target of a service; "rep_service" makes reference to a record of the Service Repository. "place", "duration" and "responsible" are analogous to the fields of the Process Summary.

The Service Summary is complemented with a Displacement and a Storage table. The Displacement table, illustrated in Table 2, describes an itinerary for each instance of a transportation service. "Id_service" is the related service identifier. "Description" describes the service carried out. "Itinerary" stores the trajectory of that transportation service: triples ($x_i$ $y_i$ $t_i$) where ($x_i$ $y_i$) are geographical coordinates and $t_i$ the time. "Company" makes reference to an enterprise, whose description is stored in the Participant Repository. "Regulation" contains references to the norms or conditions that are followed by the service in this itinerary.

Table 2: Displacement Table.

| Displacement | | | | |
|---|---|---|---|---|
| **id_service** | **Description** | **Itinerary** | **Company** | **Regulation** |
| Stransp1 | milk transport from producer1 to the dairy1 | $\{(x_1\ y_1\ t_1),(x_2\ y_2\ t_2)\}$ | Part_Transp1 | R1 |
| Stransp2 | cheese transport from dairy1 to the warehouse1 | $\{(x_2\ y_2\ t_3),(x_3\ y_3\ t_4)$ | Part_Transp2 | R1, R2 |
| Stransp3 | cheese transport from warehouse1 to the supermarket1 | $\{(x_3\ y_3\ t_5),(x_3\ y_3\ t_6)\}$ | Part_Transp3 | R3 |

The Storage table, see Table 3, contains details on instances of storage services. The field "Id_serv" refers to the service. "Company" makes reference to an enterprise that is carrying out the storage. "Regulation" contains references to the norms or conditions followed while performing this storage service.

Table 3: Storage Table.

| Storage | | |
|---|---|---|
| id_serv | company | regulation |
| St1 | Part_wh1 | id_r1 |
| St2 | Part_sm1 | id_r4 |

## 3.3 Repositories

Our model relies on six types of distributed repositories: Product, Process, Service, Regulation, Participant and Summary Repositories.

Product Repositories store generic descriptions of products that are used or generated along the supply chain. A product can belong to distinct chains. For example, the product "Minas Cheese" has a unique generic description for all chains which is instantiated every time a product of this type is produced.

Process Repositories store descriptions of the processes that generate or transform products along the chain. These processes are stored as workflows and define the sequence of activities needed for the production of a specific product. Table 4 illustrates the records stored in the repository for the example of Figure 1. The first row concerns a cow milking process and the second shows details of cheese production in the *Dairy*.

Table 4: Process Repository.

| Process Repository | | |
|---|---|---|
| id_repproc | name | workflow |
| MilkProdProc | Milk Production Process | Cows → Wash cows → Milk cows → Store → ProdMilk |
| CheeseProdProc | Cheese Production Process | ProdMilk → Produce mass → Transport → Place in mold → ProdCheese |

Service Repositories store the description of services – Transportation and Storage – that are provided by a chain's participants. Participant Repositories store descriptive data on the chain's participants - e.g, information on a transportation or a production company. Regulation Repositories contain the regulations for quality control in processes and services. Summary Repositories store summary records previously described and are of three kinds: Product, Process and Service.

## 4 ARCHITECTURE

This section shows the architecture (Figure 5) that supports our traceability model. The figure is composed of three layers: repository, summary, and summary manager layers. All layer components are encapsulated by Web services.

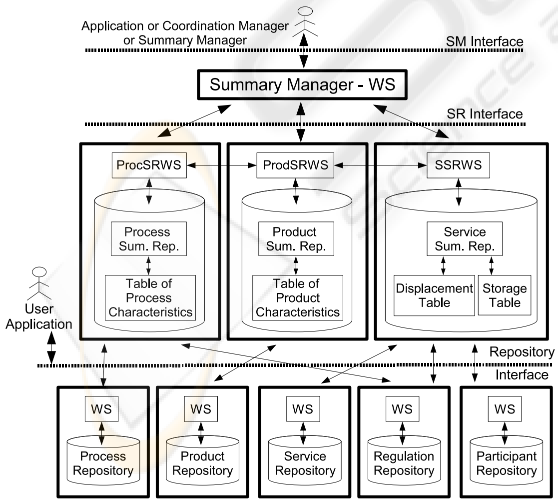


Figure 5: Proposed Architecture.

Traceability processing involves queries and updates to Repositories. It is performed via external requests to the Summary Manager Web Service (SMWS). This service provides an interface (*SM interface*) that allows client applications (a Summary Manager, a Coordination Manager or a specific software) to interact with the Summary Repositories. Through this service, a client application can request queries or update the summaries and related tables with information generated along the supply chain. Several SMWS may exist, for a fully distributed summary management scenario. This service forwards the appropriate requests (queries and/or updates) to the summary layer, which may need to forward part of these requests to the repository layer. Repositories in this layer may also be updated or queried directly by external applications.

The figure separates summary concerns from the rest – thus, Summary Repositories are treated apart. The bottom layer encapsulates access to five Repositories via five distinct Web services, that respectively receive requests to Process, Product, Service, Regulation and Participant Repositories.

The middle layer is composed of three Web services, ProcSRWS, ProdSRWS and SSRWS, respectively encapsulating access to Process, Product and Service Summary Repositories and associated tables. The figure shows that the interface of ProdSRWS accepts requests from the Summary Manager Web service, and may interact with SSRWS and ProcSRWS. The Web Service of the Product Summary Repository is a central piece in traceability processing. The other two Summary Repository services (ProcSRWS and SSRWS) cannot interact with each other. The reason for these design decisions is based on the fact that Product Summaries centralize event recording – thus, Process and Service Summaries are to some extent dependent on Product Summaries.

Though the figure shows only one repository of each kind, repositories are in fact distributed. In this case, we assume that the figure elements are replicated for each site. Distribution does not mean that all elements of the figure appear at every site. Rather, it provides a logical overview of communication among Web services and application. Moreover, traceability management must always depend on the Summary Manager service (SMWS). Thus, our distributed architecture relies on several SMWS, one for each site containing summary data. These services interact with each other and manage the access to related Product, Process and Service Repository services.

Distribution also raises issues such as access authorization, replication, and integrity control, which are beyond the scope of this paper. For our purposes, it is enough to state that interactions mediated via a given Summary Manager service must conform to network domain constraints. We say that two Summary Repositories belong to the same domain when they have mutual access authorization. The interaction between repositories in distinct domains is accomplished through Summary Managers.

Let us now illustrate Web service interactions, starting with a non distributed scenario. Consider a request R1 made to service SMWS1 to know when a given process finished producing product instance Cheese1. This is executed as follows. SMWS1 is "responsible" for interacting with three summary services – ProcSRWS1, ProdSRWS1 and SSRWS1. It sends a request to ProdSRWS1 with input parameters "id_product = Cheese1" and "event = creation" (see Figure 2), requesting as output "production_date" (see Figure 3). First, ProdSRWS1 finds out which process instance was responsible for creating Cheese1 (suppose "process_id = pr_ch1"); next, ProdSRWS1 will request from ProcSRWS1 the production date, of "pr_ch1", the desired date for R1. Repository updates can also be processed in a similar way.

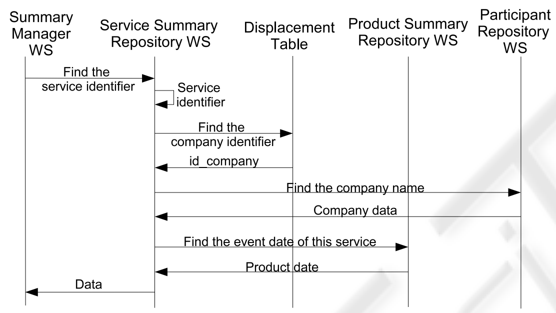


Figure 6: Example: activity sequence.

In a second example (Figure 6), a user application requests the following query from the SMWS: find the company name that carried out the transportation service of product P from dairy ($x_1y_1$) to warehouse ($x_2y_2$) and the date of this transport event. The SMWS interacts with the Service Summary Repository Web service (SSRWS) which finds the service identifier. Within this service, the company is found via the displacement table. Given the company identifier, the SSRWS poses a request to the Participant Repository WS to find the company name. To find the event date of the service, the SSRWS accesses ProdSRWS using the product and service identifiers. Date and company name are forwarded to the SMWS and returned to the user application.

Our examples concern a centralized scenario. In the distributed case, requests will start at some SMWS – say, SMWS1 – that will interact with "its" Summary services for information. If this cannot be provided, SMWS1 will broadcast the request to other SMWS which behave independently as described in the centralized case. Another possibility to avoid such broadcast overheads would be to replicate some key repository information (e.g., basic product and service data).

The use of Web Services for the communication between the Summary Manager and the many repositories facilitates the management of traceability data and provides interoperability for distributed and heterogeneous environments.

# 5 CONCLUSIONS

Supply chains present many research challenges in Computing. This paper proposed a traceability model for agricultural supply chains. It allows to follow the life of a product at distinct granularity levels, through distributed summaries and repositories. The model proposed was evaluated for different chains for specific scenarios.

This paper differs from most related work by the level of detail concerning storage aspects and the use of spatial data in repositories and summaries, increasing the number of traceable entities and facts. Moreover, it provides a Web service based infrastructure to allow interactions among repositories. The use of Web services increases flexibility in chain event management and facilitates traceability processing.

Ongoing work includes the use of real chain data, and service implementation. We adopted XML standards and services are being implemented using the Java language and PostGIS database.

# ACKNOWLEDGEMENTS

This work was partially financed by FAPESP, CNPq, as well as WebMaps and AgroFlow CNPq projects.

# REFERENCES

Alonso, G. et al. (2004). *Web Services: Concepts Architectures and Applications*. Springer, Germany.

Bacarin, E., Medeiros, C. B., and Madeira, E. R. M. (2004). A Collaborative Model for Agricultural Supply Chains. In *CoopIS/DOA/ODBASE 2004, LNCS 3290*, pages 319–336.

Bello, L. L., Mirabella, O., and Torrisi, N. (2004). Modelling and evaluating traceability systems in food manufacturing chains. In *Proc. of the 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 173–179.

Cimino, M. G. C. A., Lazzerini, B., Marcelloni, F., and Tomasi, A. (2005). Cerere: an information system supporting traceability in the food supply chain. In *Proc. of the Seventh IEEE International Conference on E-Commerce Technology Workshops*, pages 90–98.

Debord, M., Viau, A., Tychon, B., Oger, R., and Danet, V. (2001). Geotraceability: an innovative concept for the qualification of crop production. Available at: http://www.geotraceagri.net/doc/GeoTraceAgri_Final-report_EN.pdf/.

Guiffrida, A. L. and Nagi, R. (2006). Cost characterizations of supply chain delivery performance. *International Journal of Production Economics*, 102(1):22–36.

Hollingsworth, D. (1995). Workflow management coalition - the workflow reference model. In *The Workflow Management Coalition Specification*.

Kumar, K. (2001). Technology for supporting supply chain management. *Comm. of the ACM*, 44(6):58–61.

Min, H. and Zhou, G. (2002). Supply chain modeling: past, present and future. *Computer & Industrial Engineering*, 43:231–249.

Nagappan, R., Skoczylas, R., and Sriganesh, R. P. (2003). *Developing Web Services. Architecting and Developing Secure Web Services Using Java*. Wiley, USA.

Opara, L. U. (2003). Traceability in agriculture and food supply chain: A review of basic concepts, technologic implications, qne future prospects. *Food, Agriculture and Environment*, 1(1):101–106.

Stock, J. R. (2004). *Food Supply Chain Management*, chapter 14 - The US Food Supply Chain, pages 211–220. Blackwell Publishing, United Kingdon.

Thomas, D. J. and Griffin, P. M. (1996). Coordinated supply chain management. *European Journal of Operational Research*, 94(1):1–15.