56

DESTION – A DYNAMIC LINK BETWEEN EMPLOYERS AND THEIR EMPLOYEE BENEFIT PROVIDERS

Simon Field*, Gert Dazler+
*Matching Systems Ltd., SWITZERLAND
spf@matchingsystems.com
+EBZ bv., THE NETHERLANDS
gdazler@destion.nl

This paper describes the Destion system, developed by EBZ to create dynamic links between major employers in The Netherlands and their employee benefit providers (typically financial services providers). Changes in employee status are passed electronically by the employer's salary system to the Destion system, operated by EBZ. The Destion system dynamically determines which employee benefit providers should be informed of changes for each employee, and configures outgoing messages tailored for each destination provider back-end system, facilitating straight-through-processing of information between many employers and many employee benefit providers.

1. INTRODUCTION

This paper presents a case study of the design of a system that facilitates automated interaction among members of a virtual enterprise - the Destion system. It was developed in just a few months in early 2003, and is already in production use in The Netherlands. Its initial deployment links together major employers with various financial employee benefit providers (such as pension fund and savings providers). It is used to notify benefit providers of material changes to employee circumstances, such as changes of salary, marital status or address.

Complexity arises from the fact that each employer may deal with multiple benefit providers, and that individual employees can choose for themselves which benefits they take. As a result, each employee may have relationships with a variety of different benefit providers.

The system replaces ad-hoc arrangements between individual employers and their employee benefit providers, resulting in substantial cost savings and efficiency improvements as a consequence of the automated flow of information from the employers' salary systems direct to the back-office systems of the benefit providers, via the Destion system.

L. M. Camarinha-Matos et al. (eds.), *Processes and Foundations for Virtual Organizations*

2. BUSINESS PERSPECTIVE

Employers are confronted with many administrative processes that arise from their relationships with their employees. Examples include payroll processes and processes imposed on employers through legislation. One common consequence is the need to communicate changes in employee data to private and public organisations. Such messages might relate to changes to an employee's salary, the hiring and firing of employees, sickness reports, and family related changes (e.g. marriage). Employers have to send these messages to a variety of agencies, including employee benefit providers, insurance companies, health and safety at work agencies (ARBO-diensten in The Netherlands) and tax authorities.

Today, these processes are characterised by many incorrect and incomplete messages. Often, this communication is on paper, requiring the receiving organisation to re-key the data for submission to their administrative systems. Additionally, there is little transparency within the chain of information exchanges, and little efficiency in the administrative processes. These problems arise because of the absence of a common information and communication standard. Until now, competition and rivalry between organisations, and internal politics within organisations, have prevented the emergence of a common standard.

Destion facilitates the exchange of such messages (known as 'mutation messages') between employers and other parties such as employee benefits providers. By using recent Internet technologies such as xml (Bray, 2000), intelligent agents and business rules and a component-based infrastructure (J2EE, 2002), Destion enables standardised, efficient, and automated communication of messages between different entities within the value chain. The system can be characterised as an inter-organisational system (IOS) because it links the ERP systems of employers with the back-office systems of other organisations (such as employee benefits providers).

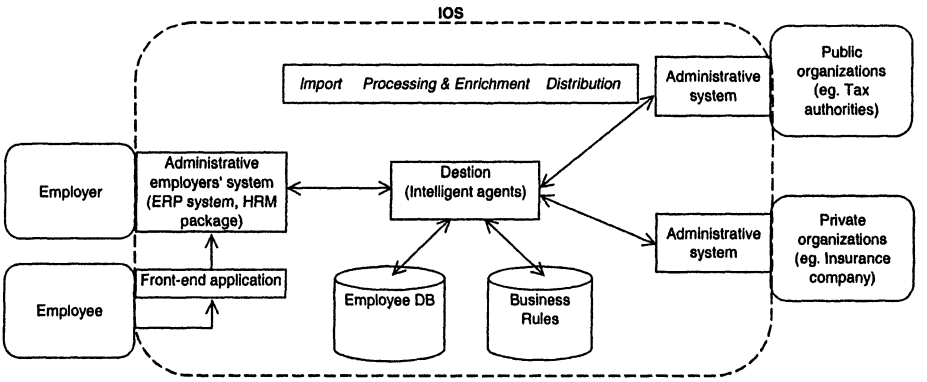

Figure 1 – global view of Destion

Figure 1 gives a global view of the concept behind the Destion system. A mutation message is generated by using a front-end data entry application or extracted from the administrative back-office system of the employer. The xml

message is sent to Destion. Destion performs several checks (authorisation, authentication) and validates the message with the corresponding xml schema. Based on existing data in the employee database, Destion enriches the message. The enriched message is sent to the matching component where it is validated and interpreted based on business rules. If a message seems incomplete, Destion notifies the employer, and requests additional information. A complete message is decomposed into several outgoing xml messages, one for each receiving organisation. These messages are, if necessary, converted to a specific format preferred by the receiver. This is typically xml, but it is possible to receive the message by email or a system specific format (e.g. ASCII). Destion uses several standard components for user provisioning and billing. Bills are automatically generated from a data warehouse and sent to an SAP system.

The major business benefits arising from the use of the Destion system derive from two aspects. First, the digitalisation of mutation messages offers opportunities to optimise operational efficiency. The processing of the message can be more easily automated. Second, the intelligence built into the Destion system results in messages of higher quality. Moreover, organisations can handle a larger quantity of messages and reduce the delay in information exchange, thereby improving customer service.

The first phase of the Destion project involves the exchange of messages relating to employee benefits, such as pension products and services provided by health and safety at work agencies. Because Destion can deliver messages in the format desired by the receiver, the receiver can process messages automatically. This leads to considerable savings for the receiver, compared to the manual processing that has taken place until now. These savings can be passed on to the employers, and some life insurers are already offering lower prices if they are able to process mutation messages automatically. Health and safety agencies have adopted the same strategy. Today they receive many thousands of sickness reports by fax. Employers that deliver such reports in a format that can be processed automatically are offered a discount that can exceed 10%.

3. SYSTEM OVERVIEW

3.1 System Setup

Employers and their benefit providers must first register with EBZ, so that data and message formats can be agreed in advance. The Destion system has been designed for maximum flexibility. Receipt of mutation data from an employer is expected as an xml file, the precise format of which can be varied from employer to employer to suit the limitations of their existing systems.

Outgoing messages can be sent via the AIMS messaging system from ABZ (ABZ, 2003), to which most benefit providers already subscribe. Alternative message formats are also supported, including email. The format and content of outgoing messages can be tailored for each mutation event to suit the requirements of each provider's back-office administrative system. Specifying the business rules and format information for each employee benefit product is a simple task,

involving the creation and submission of a set of xml files. See Section 4.2 below for more details.

Once an employer and its benefit providers have registered with EBZ, the Destion database must be loaded with employee benefit data. This comes from both the employer and the benefit providers, and enables EBZ to identify which employees have subscribed to which benefit products from which providers.

3.2 Regular Operation

Regular operation of the Destion system involves the periodic receipt of employee changes (or mutations) from the employer's administrative systems. This triggers the need to select and configure appropriate outgoing messages to the various benefit providers that need to be notified of the changes to employee status. The system has been implemented in Java, with an Oracle database, and deployed in a clustered BEA WebLogic application server environment.

The core matching functionality is described in Section 4 below, and the overall process is described in detail in Section 4.4. Administrative aspects of the system, such as billing, audit logging and the like, are considered outside the scope of this paper, and have been excluded for the sake of clarity.

4. THE MATCHING COMPONENT

4.1 The Matching Systems Engine

At the heart of the Destion system lays the Matching Systems Engine, a generic matching engine component developed by Matching Systems Ltd. (MSL, 2003). The Matching Systems Engine is a commercial spin-off derived from the work of the e-business Solutions Group at IBM's Zurich Research Laboratory (Hoffner, 2000 and Field, 2002).

The Matching Systems Engine holds a set of advertisements that represent entities that may be matched. Each advertisement consists of a set of descriptive properties and a set of business rules that determine the circumstances in which that advertisement will be matched with an incoming query. The descriptive properties can be either static (i.e. the same value irrespective of the incoming query), or dynamic in which case code is included that will dynamically determine the value in the light of the properties of an incoming query.

An incoming query triggers the matching process. Each query consists of a set of descriptive properties that may be referenced by the advertisement business rules or dynamic property code. The result of a query is a set of matching offers – each offer representing the resolved set of properties of an advertisement whose business rules matched with the incoming query.

4.2 Message Matching

For the Destion System, the Matching Systems Engine dynamically determines which employee benefit providers should be informed of changes (also known as

mutations), and configures outgoing messages tailored for each destination provider back-end system. Each potential outgoing message is represented as an advertisement. Its properties describe the information content of the outgoing message, mostly dynamic content to be obtained from the incoming query properties. The message properties are also used to indicate the preferred format expected by the message recipient. Each advertisement also contains business rules that determine the circumstances in which this particular outgoing message should be selected (see Figure 2 for an extract from a sample message advertisement).

```
<?xml version="1.0" encoding="utf-8"?>
<Advertisement>
   <Rule><![CDATA[
         return exists("salarismutatie");
   ]]></Rule>
   <Rule><![CDATA[
         return (myRecord("polis") != null);
   ]]></Rule>
   <type>message</type>
   <verzekeraar>Delta Lloyd</verzekeraar>
   <mutatiesoort>salarismutatie</mutatiesoort>
   <ingangsdatum_mutatie dynamic="true"><![CDATA[
      return yourString("ingangsdatum_mutatie",
         yourRecord("salarismutatie"));
   ]]></ingangsdatum_mutatie>
   <relatienummer dynamic="true"><![CDATA[
      return yourString("relatienummer",
         yourRecord("werkgever"));
   ]]></relatienummer>
   <polis type="Record" local="true" dynamic="true"><![CDATA[
      return nl.abz.destion.mse.Util.findPolicy(
         myString("verzekeraar"),
         myString("product"),
         yourSequence("polissen"));
   ]]></polis>
   •
   •
   •
```

Figure 2 – sample message advertisement (extract)

The system can be very easily maintained and extended. New message advertisements can be added to the system simply by creating an appropriate xml document and submitting it to the Matching Systems Engine.

4.3 Error Matching

The same Matching Systems Engine is also used to validate the incoming information. In this case, each advertisement represents an error situation – the rules indicating the circumstances when an error is detected, and the properties indicating the appropriate action to be taken when the error is found.

The system has been set up so that on receipt of the query it will first attempt to match errors. If it finds some, it returns the details of the errors found. If no errors are found, the system then seeks matching message advertisements.

4.4 The Matching Process

Figure 3 illustrates the runtime roles that the Matching Systems Engine performs for the Destion system and its relationship with other components in the overall system.

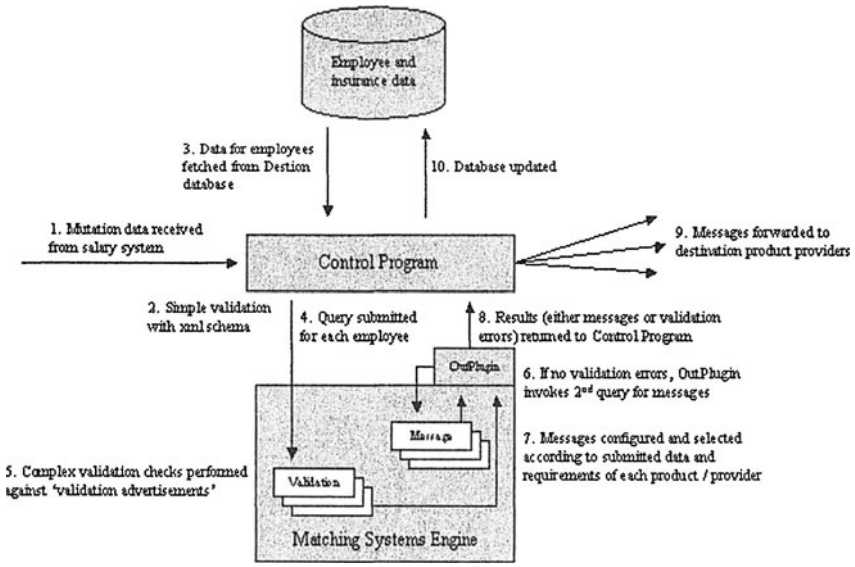

Figure 3 – the matching process

1. Data is received from the salary system or the front-end application of the employer.
2. The incoming data is validated at field level against an xml schema. This verifies that required fields are present, with appropriate data types and values. It does not perform more sophisticated validation, such as checking the consistency of one field against that of another (due to the limitations of schema validation).
3. Data from the Destion Database is obtained for each employee for whom mutation data has been received.
4. A query is submitted to the Matching Systems Engine for each employee. The query data, consisting of the combined mutation data for that employee with his or her employee and insurance data from the Destion Database, is passed to the Matching Systems Engine as a java HashMap object.
5. The Matching Systems Engine matches *validation advertisements* against the query data. Each advertisement represents an error condition, potentially comparing the consistency of data across multiple fields. The result of this

stage of the matching process is a (possibly empty) set of error conditions that apply to the query data that was submitted.

6. A simple OutPlugin determines whether any error conditions have been found. If so, these are returned to the Control Program, and the matching process is complete (as there are errors, there are no messages to be sent to product providers). If there are no error conditions, the OutPlugin sends the query data back into the Matching Systems Engine for a further query, this time seeking *message advertisements.*
7. The Matching Systems Engine matches *message advertisements* against the query data. Each advertisement represents a possible output message for a particular product. As it is possible for the query data to contain more than one mutation, it is necessary for each possible mutation to have its own *message advertisement.* A match with a *message advertisements* results in the creation of a *message offer,* containing the information content of the message that has been automatically selected and configured during the matching process. Whilst there are alternative ways of representing messages within the Matching Systems Engine, this solution offers maximum flexibility, allowing business rules and message content to vary by mutation, provider or individual product, or any combination of these. A further advantage of this representation is that there is a simple one-to-one mapping between *message offers* and output messages to be sent to product providers.
8. The OutPlugin returns the resulting offers to the calling Control Program, in the form of a java Vector of java HashMaps. The HashMaps will either all be error conditions, in which case the Control Program passes them to the salary system and the employer, or *message offers,* in which case the Control Program can construct the appropriate outgoing message for the destination product provider.
9. The Control Program sends the outgoing messages, built from the *message offers* returned by the Matching Systems Engine.
10. The Destion Database is updated with information that was received in the mutation data, ensuring that the Destion Database is up to date for that employee.

5. FUTURE WORK

To date the main focus of the project has been on the receiver of messages – in this case a Dutch life insurance company, and on the savings that they can achieve by using the system. In the future more attention will be given to the other side – the employers where the messages originate. The costs for an employer to provide information to other parties are very high, especially if each party asks for different information in different formats. As mentioned in Section 2 above, receiving parties can be private or public organisations. Public organisations in particular define strict requirements for messages that must be met by employers. They expect information in their format, at a time of their asking. There is also a lack of consistent definitions. For example, the tax authorities define salary to be composed of different elements than it is for social security authorities. The Dutch government has recognised this complexity and is preparing legislation, WALVIS, to reduce the

burden of administrative processes (Niessen, 2002). The goal of WALVIS is to come to an unambiguous terminology in the area of salaries, and as a result reduce administrative costs for both employers and government. Other government regulations, for example in the area of social security, might also affect communication between private and public organisations.

In the future, Destion will be extended by linking more organisations to the system and adding new functionality. The purpose is to enable employers to send only one message per employee-triggered event (e.g. marriage, change in salary, change of address). The message will be sent to Destion, which deals automatically with all resulting information exchanges. Employers have to be informed what information has been sent on their behalf, and to whom. Therefore, Destion must be extended with a new component that provides employers with an overview of messages that have been sent and their corresponding receivers. In such an ideal situation, all parties will realise operational savings and profit from increased transparency across the supply chain.

The system is already being used by a number of early adopter providers – a major pension provider and a leading health insurer. The generic nature of the solution has triggered the further development of a hub for absenteeism that will coordinate the administrative processes for both employers and the relevant authorities. Currently, the settlement of insurance claims relating to this is particularly inefficient, due to the high cost of processing unstructured messages. In addition to the distribution of information from employers to relevant authorities, the hub can provide claim verification and fraud signalling services for insurance companies, and help automate a digital claims service between employers and insurance companies.

6. REFERENCES

1. Bray T., Paoli J., Sperberg-McQueen C. M., Maler E. "Extensible Markup Language (XML) 1.0 (Second Edition), W3C, 2000.
2. J2EE, http://java.sun.com/j2ee/overview.html, 2002.
3. Matching Systems Limited, http://www.matchingsystems.com, 2003.
4. Hoffner Y., Facciorusso C., Field S., Schade A. "Distribution issues in the design and implementation of a virtual market place." Computer Networks 2000; 32:717-30.
5. Field S., Hoffner Y. "In Search of the Right Partner", Collaborative Business Ecosystems and Virtual Enterprises, ed. Luis M. Camarinha-Matos, Kluwer Academic Publishers, 2002.
6. ABZ bv. http://www.abz.nl/indexeng.html, 2003.
7. Wierda, F.., Developing Interorganizational Information Systems, Proefschrift (Thesis) Delft, 1991
8. Niessen, R., WALVIS vermindert administratieve lasten werkgevers, Forfaitair 2002, 127:24-28