A Data Space System for the Criminal Justice Chain

Jan van Dijk[1], Sunil Choenni[1], Erik Leertouwer[1],
Marco Spruit[2], and Sjaak Brinkkemper[2]

[1] Dutch Ministry of Security & Justice, Research & Documentation Centre (WODC),
Turfmarkt 147, 2511 DP The Hague, The Netherlands
{j.j.van.dijk,r.choenni,e.c.leertouwer}@minvenj.nl
[2] Department of Information and Computing Sciences, Utrecht University
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
{m.r.spruit,s.brinkkemper}@uu.nl

Abstract. In this paper we present the concepts and implementation of a data space system for the management of data from heterogeneous sources in the criminal justice field. Our system exploits domain knowledge in the field of justice to streamline and to relate the content of different databases in the chain. In our system, domain knowledge is encoded in a space manager layer. Furthermore, in this layer it is decided which databases should be used to answer a query. This decision is taken on the basis of the encoded domain knowledge, and the content of the databases and its quality.

Keywords: data space system, criminal justice chain, domain knowledge, chain management.

1 Introduction

While in many fields data from different databases are related via automated data integration concepts and techniques – such as schema mapping or tuple mapping approaches (see amongst others [1-3]), data in the field of justice are manually or semi-manually related. The fact that data are related in a (semi-)manual manner in this specific field has to do with the complexity that is required in understanding and interpretating the data ([4-6]). In our case relating data means to bring (aggregated) data together that pertain to a same real world entity and define how the data should be interpreted in the context of other real world entities [7]. Furthermore, a concept of primary/foreign key relation between different databases is often lacking. If such a relation does exist, privacy law and regulations may not allow us to use it. Therefore, relating data from different information systems (semi-)manually is an error-prone and tedious process.

In this paper, we present the concepts and implementation of a data space system for the management of data from heterogeneous databases in the criminal justice field. The term data space system was coined in [8], as an agile form of data integration using a "pay-as-you-go" approach. Meanwhile the concepts in [8] have been elaborated in different directions ([6], [9-10]) and several prototypes have been developed

R. Meersman et al. (Eds.): OTM 2013, LNCS 8185, pp. 755–763, 2013.

[11-15]. Kalidien et al. [6] come up with a conceptual architecture for data space systems, while Hedeler et al. [10] present a functional model for data space management systems which can be seen as a first attempt to formalize data space operators. The efforts in [11-12] combine personal information, [13-14] focus on the combination of data from domain independent sources, and [15] supports semi-automatic data integration of data sources in the domain of life science. We have built upon the concepts introduced in [8] and [6].

The main goal of our system is to give a more complete look into the flows through the criminal justice chain so that policy makers can identify potential capacity problems in an early stage. Also, an important requirement of such a system is that it should be able to combine all kinds of data in the justice field such as registered data from chain partners, forecasts, and safety survey data.

The importance of managing data between different databases is in some sense recognized in the field of federated databases and data warehouses. In a data warehouse, data from different sources that pertain to a single entity is transformed and loaded. Entities that apparently pertain to the same real-world object are linked together by means of primary and foreign keys. In [16], it has been pointed out that non-key attribute values are often inconsistent, while this is less often the case for key attribute values. In [6] it is argued that developing a data warehouse is a costly effort, and the development of a data space system is not only cheaper but may be also more effective for heterogeneous data management, e.g. in the field of justice. Although our system is tailored for the Dutch criminal system, we note the notions and concepts of data space systems (see Section 2) may be used in fields where relationships between different databases are of crucial importance.

2 Defining Data Space System Concepts

While database management systems provide an extensive set of tools to manage data within a database, tools to manage data between different databases are lacking. There is a practical need for these tools by organizations that are embedded in a chain and especially organizations that are in charge of the control of the chain. Insight into the flows through the chain is required to identify bottlenecks and improvement of the chain. For these reasons, concepts, tools and techniques need to be developed to manage heterogeneous data from the different systems in the chain. This requires an extension of the concepts and techniques of database design. In general, database design primarily focuses on those data required to serve the information needs of a selected group of users, e.g. a department in an organization. In database design it is not a common practice to establish relationships between different databases. A database is considered an independent and autonomous unit, in which the relationships with other databases are neglected and therefore not specified. It is exactly this property for which a data space system primarily differs from a database system [6]. In a data space system, we distinguish three layers, see also Fig. 1: an interface layer

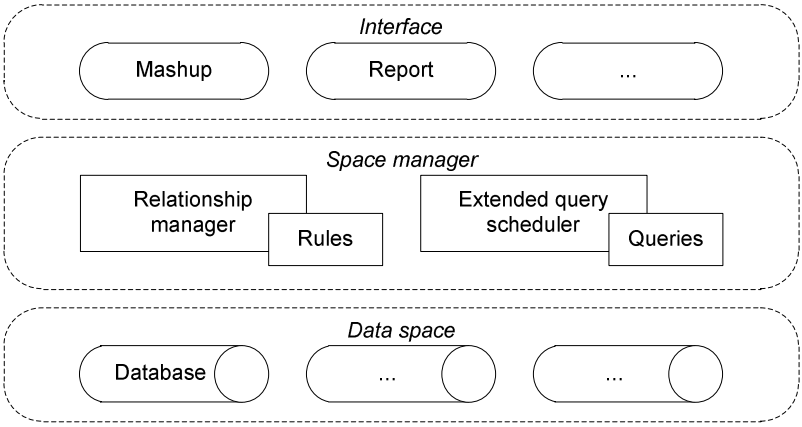

Fig. 1. Conceptual model of a data space system (Source: [6])

with querying and presentation functionality for end users; a space manager layer, containing a relationship manager with knowledge about the data in the data space and a query scheduler module; and a data space layer, consisting of a collection of databases.

In the data space layer, no constraints are put on the type of databases. The relation between different databases is encoded in the relationship manager of the space manager layer. The term "relation between databases" is defined in a broad sense. Two databases are related if they contain data about a same real-life entity or they contain data that may be used to establish a link between different entity types. For example, a criminal case brought to court leads to a verdict. Therefore, the number of cases brought to court and the number of cases with a verdict are related. So, the number of verdicts is less or equal to the criminal cases brought to court. This is a relationship between the database that records verdicts and the database that records criminal cases. The extended query scheduler module is used to determine which attributes are used to answer a query. In contrary to a query optimizer, the choice for the attributes is not based on trade-off options of (cpu and/or disk access) cost [17] but rather on the quality of the attributes [18]. The quality of the attributes is rated by domain experts and important factors that influence the rate are completeness and timeliness of the attribute data [19].

The interface layer is primarily meant to communicate with end users. Analogous to the terminology in database systems, the interface and the space manager layer may be regarded as data space management system. These layers adequately handle a query from an end user. Analogous to ANSI-SPARC architecture for the development of databases, the layers in the data space system are independent. For example, we allow making modifications at the space manager, while leaving the data space as it is. This property of independency contributes to the flexibility and extensibility of systems based on the data space system concepts.

3 Designing a Data Space System

Since the nature of the data in the criminal justice chain differs from administrative data ([5-6]), we summarize the characteristics of the data and relationships in the first section. In Sections 3.2 and 3.3, we discuss our design choices.

3.1 Data and Relationship Characteristics

Similar properties of a real-world event may be stored differently in the information systems of organizations involved in the criminal justice chain. For example, an event may be labeled by the police as a burglary with violence. However, the public prosecutor may come to the conclusion that violence will be hard to prove, therefore the same event is labeled as a burglary. Also, chain partners may use their own definition for seemingly similar data, depending on their core processes and management goals. Furthermore, chain partners register their data at different points in time, namely at the time when their involvement is required, and also at different precision levels, since detailed information relevant to one chain partner may not be relevant to another. For example, courts register the number of sentences imposed while the executing organizations register the number of sentences executed (some convicts cannot be found or die before the sentence can be executed). Courts register the type of crime because it is relevant for the sentence, but executing organizations may not record it as it has no relevance for the execution. Finally, we note that organizations at the beginning of the chain, such as the police, and at the end of the chain (the executing organizations) are primarily interested in data that pertain to individuals, while the remainder of the chain partners is rather interested in data about a case than an individual. In our system we take advantage of above-mentioned properties to relate (entities in) different databases.

To ensure the quality of the data, sets of rules are implemented in a separate relationship module. We distinguish between two types of relationships. The first type deals with missing data. If the value of an attribute is (temporarily) unavailable, for example due to technical problems, it may be replaced with the value of a similar attribute that is measured at a slightly earlier or later point in time, but more or less covers the same notion. This is called imputation. The imputed value may come from either the same or a different source. For instance, if the number of summonses is unknown, but we do know when the first court session took place, we may insert the value of the number of first sessions as an approximation for the number of summonses. In most cases these two attributes are highly correlated, because each summons eventually has to lead to a court session. Defining such relationships can help us to choose which data need to be selected when the preferred data are not available.

The second type of relationship deals with differences between seemingly similar but actually very different data. Different chain partners may label their data using a similar or even the same names, but because these data are measured on different points in time and based on their own criteria, the outcomes for these data will vary. Knowing the relationship will help us compare the data correctly. This problem is

generally known as a semantic problem, and is difficult to detect [1]. To illustrate the second type of relationship, consider the number of community services agreed with the public prosecutor. These data come from two chain partners, namely the public prosecutor and the execution organization. Thus, the relationship rule for these data is that the number of community services registered by the executing organization should be lower or equal than the number registered by the public prosecutor, because suspects may die before execution or cannot be found. Historical data confirm this rule. The data are called stable when they meet this rule.

3.2 Relationship Rules

To construct relationship rules, domain knowledge is indispensable, and historical data can be used to confirm them. To apply the rules into the current data, there are two major practical challenges that need to be considered:

- The current data at hand may contain temporary missing data: this is understandable as organizations have different levels of capacity to handle data request and delivery, and thus, not all of them will be able to provide all data on time.
- The observed data may depend on other, possibly temporary factors: this is more difficult to detect especially if it pertains a change in policy. For instance, until recently the police had a so-called 'bonnenquota' (fine quota). As long as a high fine quota should be met, the police will quite easily giving out fines.

Considering these factors, it is suggested to define relationship rules in such a way that they will allow some violations. In other words, the rules should not be implemented in a strict manner (such rules are usually called soft rules).

In our application, we define the rule as a comparison between the ratio of the related data between two organizations and their average ratios. The previous data values are used to determine the average ratio. To stay as close as possible to the current situation, it seems justified not to use too much historical data. In our case, we apply the relationship rules to quarterly data and the average is calculated based on a series of quarterly data in the past two years. Violations of the rules in certain range of values will be allowed. The rules can be formally defined as follows.

$$(1-\beta)*\nabla_i \le \frac{x_i^{(j),t}}{x_i^{(k),t}} \le (1+\beta)*\nabla_i, \text{ where} \tag{1}$$

i = the related variable (data)

j,k = data sources of this variable

$\frac{x_i^{(j),t}}{x_i^{(k),t}}$ = the ratio of variable i from source j and variable i from source k, at time period t

∇_i = the average ratios of variable i from source j and variable i from source k, calculated at time periods t-1, t-2, ...

β = margin for violation, $\beta \in (0,1]$

Using a margin of 10% (ß=0.1) means we allow the ratio of the present data to fall between 10% less than the average and 10% more than the average. Deciding which margin value will be reasonable requires some trial and error. Alternatively, one can observe a standard deviation calculated from historical data. In our case, most of the variables have a standard deviation around 0.1, but for some variables the standard deviation can vary up to 0.15. If we choose the largest standard deviation, it means we underestimate possible violations of the rules (or in other words: we allow the largest violation possible).

3.3 Variables

We have organized variables in hierarchical trees. For example, the database holds the following variables $v_1..v_6$: {C-Cases-M-Verdict-Acquittal, C-Cases-M-Verdict-Conviction, C-Cases-C-Verdict-Acquittal, C-Cases-C-Verdict-Conviction, C-Cases-M-Verdict, C-Cases-C-Verdict}. Fig. 2 depicts the tree for these variables. The variable trees are constructed bottom-up: the lowest (most specific) variables in the tree are generated directly from attributes, and upper variables are generated from lower variables.

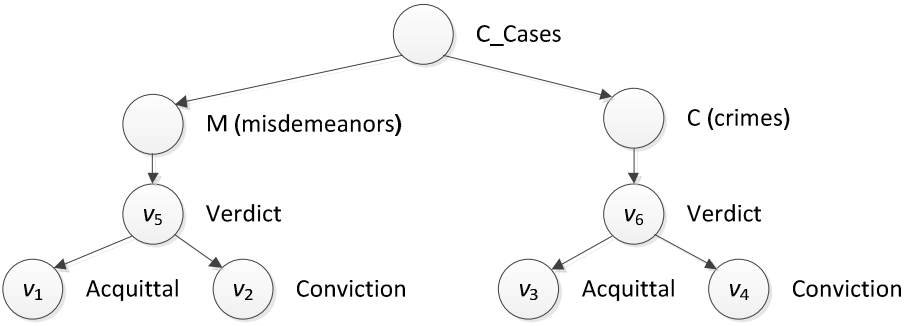

Fig. 2. Tree of variables that pertain to court cases

Variables are constructed dynamically by using naming conventions in the variable names. The name of a variable is built up from different parts, distinguished by an hyphen. Each part has its own meaning. So, the variable meaning can be constructed from its parts. The first and the second part of the variables (C-Cases) refer to that all variables pertain to court/trial cases. The variable v_1 pertains court cases that are misdemeanors (M) with 'acquittal' as verdict, while v_2 pertains to the same kind of cases, but with a 'conviction' as verdict. Analogous to misdemeanors, we have comparable variables v_3 and v_4 for crimes (C). The sum operator can not always be applied in a straightforward manner in a tree. This is the case whenever the attributes that are used to compute a variable are not disjunct. For instance, a verdict can have more than one 'verdict type'; an acquittal for one part of the crime, and conviction for another part. As a result, the total number of verdicts has to be extracted separately from the data space.

4 Architecture of a Data Space System

On the basis of design decisions, as discussed in Section 3, we discuss the architecture of the data space system. The architecture is presented in Fig. 3. The data space layer provides access to data sources from different organizations, which contain data of different aggregation levels. The space manager layer contains the knowledge and functionality to interpret the data, and, besides the variable database and the relationship module, consists of two auxiliary modules to create the variables. Data that are required to create a variable are extracted from the data space layer and converted to a standard format. Then, the converted data are manipulated in order to create the value of a variable and stored in the variable database. The interface layer actually presents the data to end users.

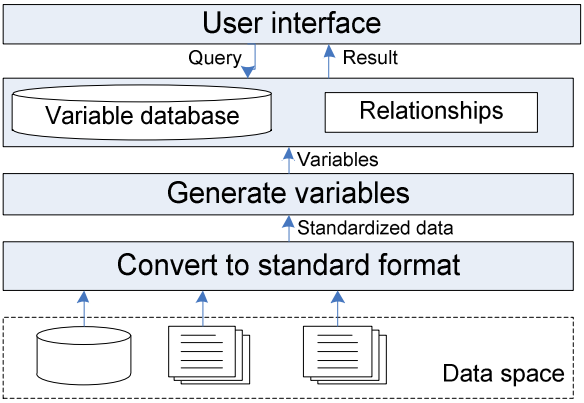

Fig. 3. Architecture of the data space system

We note that the data space layer is environment-dependent, and the interface layer is application-dependent. This means that the content of the data space is mainly determined by the environment in which the data space system is set up, whereas the way the interface is implemented depends on the requirements defined by the user. As a result, the implementation of the space manager is also application-dependent.

In this system, comparability is a key issue. Therefore, attribute data are extracted from the data space and stored in the space manager layer in a standardized format (analogous to the federated database approach [1]). On storage, attribute data is aggregated to three-month periods. The data are multi-dimensional and in this case, the dimensions are predefined: region, crime type, age, and person type (natural or legal person).

The Relationships component is a knowledge base about the variables in the data space system and their relationships. In the knowledge base the definition of variables and the relationships between variables are stored. As discussed in Section 3.3, the definition of a variable is stored in trees. Each phase in the criminal justice chain has one or more of these trees (e.g. the investigation phase has different trees for suspect and crime statistics). The knowledge base also contains the relationships rules (of the

second type) between variables. Variables are generated from the attributes using the variable trees defined in the Relationships component. When an attribute is not available for one or more periods, these periods are not generated for variables that depend on this attribute. Special variables are the 'effect variables': an effect variable entails the margin of violation of a relationship rule. All variables are stored in the variable database for use in the interface.

5 Conclusions and Further Research

While database management systems provide an extensive set of tools to manage data within a database, tools to support the management of data between different databases are lacking. For several reasons, there is a need to support the management of data from heterogeneous information systems. For example, in the field of justice this might be to gain insight into the criminal justice chain by generating management information. To support the management of data from different heterogeneous systems, we have developed a data space system tailored to the Dutch criminal justice chain.

In our system we have exploited domain knowledge in the field of justice to streamline and to relate the content of the different data sources. The domain knowledge is encoded in the space manager layer. Furthermore, in this layer it is decided which databases should be used to answer a query. This decision is taken on the basis of the content of the databases and its quality. The system proves to be successful in managing data from different information systems that are involved in the criminal justice chain. Currently, the system is used by several chain partners as well as by policy makers at the Dutch Ministry of Security and Justice to gain insight into the flows through the criminal justice chain.

To our best knowledge, our data space system is the first implementation of the data space approach targeting the criminal justice chain. A topic for further research is the generalization of the data space concept by applying it in different contexts and domains. The data space concept will be further developed with a focus on data quality aspects. Another aspect of further research is the margin of violation used in the relationship rules. Learning from this case, we believe that for a good performance the margin should be made dynamically in two ways. Each variable may have its own margin of violation, and the margin of violation can change in time. Further research is necessary to determine how this can be implemented in a reliable and maintainable way.

References

1. Sheth, A.P., Larson, J.A.: Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Computing Surveys 22(3), 183–236 (1990)
2. Cabibbo, L., Torlone, R.: Integrating heterogeneous multidimensional databases. In: Frew, J. (ed.) 17th International Conference on Scientific and Statistical Database Management, pp. 205–214. Lawrence Berkeley Laboratory, Berkeley (2005)

3. Winkler, W.E.: Matching and record linkage. In: Cox, B.G., Binder, D.A., Chinnappa, B.N. (eds.) Business Survey Methods, pp. 355–384. Wiley, New York (1995)
4. Wilkins, D., Pillaipakkamnatt, K.: The effectiveness of machine learning techniques for predicting time to case disposition. In: 6th International Conference on Artificial Intelligence and Law, pp. 106–113. ACM, New York (1997)
5. van den Braak, S., Choenni, S., Verwer, S.: Combining and Analyzing Judicial Databases. In: Custers, B., Calders, T., Schermer, B., Zarsky, T. (eds.) Discrimination & Privacy in the Information Society. SAPERE, vol. 3, pp. 191–206. Springer, Heidelberg (2013)
6. Kalidien, S.N., Choenni, R., Meijer, R.F.: Crime statistics online: potentials and challenges. In: 11th Annual International Digital Government Research Conference on Public Administration Online: Challenges and Opportunities (dg.o 2010), pp. 131–137. Digital Government Society of North America (2010)
7. Choenni, S., van Dijk, J., Leeuw, F.: Preserving privacy whilst integrating data: Applied to criminal justice. Information Polity 15(1,2), 125–138 (2010)
8. Franklin, M., Halevy, A., Maier, D.: From databases to dataspaces, a new abstraction for information management. SIGMOD Record 34(4), 27–33 (2005)
9. Hedeler, C., Belhajjame, K., Fernandes, A.A.A., Embury, S.M., Paton, N.W.: Dimensions of dataspaces. In: Sexton, A.P. (ed.) BNCOD 2009. LNCS, vol. 5588, pp. 55–66. Springer, Heidelberg (2009)
10. Hedeler, C., Fernandes, A.A.A., Belhajjame, K., Mao, L., Guo, C., Paton, N.W., Embury, S.M.: A functional model for dataspace management systems. In: Catania, B., Jain, L.C. (eds.) Advanced Query Processing. ISRL, vol. 36, pp. 305–341. Springer, Heidelberg (2012)
11. Blunschi, L., Dittrich, J.P., Girard, O.R., Karakashian, S.K., Salles, M.A.V.: A dataspace odyssey: The iMeMex personal dataspace management system. In: Proceedings of the Conference on Innovative Data Systems Research (CIDR), pp. 114–119 (2007)
12. Dong, X., Halevy, A.Y.: A platform for personal information management and integration. In: CIDR, pp. 119–130 (2005)
13. Das Sarma, A., Dong, X., Halevy, A.: Bootstrapping pay-as-you-go data integration systems. In: SIGMOD, pp. 861–874 (2008)
14. Madhavan, J., Cohen, S., Dong, X.L., Halevy, A.Y., Jeffery, S.R., Ko, D., Yu, C.: Web-scale data integration: You can afford to pay as you go. In: CIDR, pp. 342–350 (2007)
15. Leser, U., Naumann, F.: (Almost) hands-off information integration for the life sciences. In: CIDR, pp. 131–143 (2005)
16. Agarwal, S., Keller, A.M., Wiederhold, G., Saraswat, K.: Flexible Relation: An approach for integrating data from multiple, possibly inconsistent databases. In: 11th International Conference on Data Engineering, pp. 495–504. Stanford Univ., CA (1995)
17. Choenni, S., Blanken, H., Chang, T.: On the Selection of Secondary Indices in Relational Databases. Data & Knowledge Engineering 11(3), 207–233 (1995)
18. Berti-Équille, L.: Quality Awareness for Managing and Mining Data. Habilitation à diriger des recherches. L'Université de Rennes, France (2007)
19. Pipino, L.L., Lee, Y.W., Wang, R.Y.: Data quality assessment. Communications of the ACM - Supporting Community and Building Social Capital 45(4), 211–218 (2002)