Spatiotemporal Data Modeling for Disaster Management in the Netherlands

Arta Dilo
A.Dilo@tudelft.nl

Sisi Zlatanova
S.Zlatanova@tudelft.nl

Delft University of Technology,
Jaffalaan 9, 2628 BX Delft, The Netherlands

ABSTRACT

This paper presents an approach for modeling data collected during emergency response starting from a conceptual level until the implementation in a Database Management System (DBMS). This approach follows arrangements for disaster management in the Netherland. Any disaster incident is managed through processes. Each process has a well-defined objective, which realization requires certain information and often produces information. The information is mostly spatial information, and in a large part of a temporal nature. We have created new data types for the temporal and spatiotemporal data: dynamic counts, moving point, and moving region. The model is implemented in Oracle Spatial. The temporal data types are implemented using the Oracle functionality on nested tables. Visualization of spatiotemporal data is done via views that perform un-nesting of tables, and transform the data to simply spatial types. Spatiotemporal data can be used in analysis for supporting the decision making during disaster incidents.

Keywords

Disaster management, data modeling, spatiotemporal data, Oracle Spatial.

INTRODUCTION

Availability of information and fast access to the right piece of information is crucial for disaster management (DM). Information needed for disaster management consists of existing information, e.g. information for buildings, road and utility networks, and information collected during disasters, e.g. location of a disaster incident, extent and possible escalation, number of victims; we call the latest ‘operational’ information. While existing information is defined in various data models (van Oosterom and Lemmen, 2006; Quak, van Oosterom, de Vries, Bruns, Bakker, Reuvers, and Janssen, 2005; de Vries, Tijssen, Stoter, Quak, and van Oosterom, 2002), operational information is still insufficiently modeled and hardly managed. Operational information is mostly geographically related (Xu and Zlatanova, 2007), and in a large part very dynamic. Currently most of the operational data is kept as individual files (voice records, reports, video, images, measurements, sensor records) as no systematic approach is followed. Dynamic models, as far as they exist, are developed for specific types of disasters, e.g. flooding (David, Evaux, and Masson, 2006; Erlich 1996). The emergency response sector generally lacks a model for maintenance of operational data.

This paper presents an approach for modeling of operational data starting from a conceptual level until the implementation in a DBMS. The approach strictly follows the disaster management processes in the Netherlands as described in the law. Each incident in the Netherlands is managed though processes. Each process has a well-defined objective, which realization requires certain information and often produces information during its execution. Depending on the type of process, different response units, fire brigade departments, police stations, medical services, and municipalities get involved in the incident. People from these organizations perform their tasks based on assigned roles and responsibilities. Much of the information produced during disaster management is temporal, i.e. it is changing with time, and we need to keep track of changes. The structure and relations in the operational information are captured by the data model: incident and its effect, processes activated to handle it, responsible departments, persons (system users) involved in each process and their roles, measurements, etc. The data model is built from information requirement collected from fire brigade and police sectors (Snoeren 2006). New data types are created for temporal and spatiotemporal information: dynamic counts to store, e.g. number of injured; moving point for, e.g. the position of a vehicle; moving region for, e.g. gas plume. Spatiotemporal data can be further used in analysis for supporting the decision making during disaster incidents.

This paper is organized as follows. Section 2 gives overview of disaster management arrangements in the Netherlands. Section 3 presents our model for operational data, first giving an overview focusing on classes (without attributes) and their associations in Section 3.1, then going into more detail, class attributes and constraints in Section 3.2. Section 4 provides definitions of temporal and spatiotemporal types in Oracle Spatial, as well as working with spatiotemporal data. Section 5 closes the paper with conclusions and future work.

DISASTER MANAGEMENT IN THE NETHERLANDS

Legislation

Emergency response processes in the Netherlands are legally arranged within the Law for Disasters and Large Accidents (WRZO, http://wetten.overheid.nl/). The document provides definitions, describes responsibilities, manner of working, levels of emergencies, and provides classification of disasters. According to this law, the board of Mayor and Aldermen on the local level is charged with drawing up a mandatory disaster plan. In this plan, the emergency response activities and the organizational structure should be described. The organization of emergency response in the Netherlands is divided into a local level, that is the site of an incident; the regional level, emergency services are regionally organized, supporting several municipalities; the provincial level. Most emergency incidents of a minor nature are responded at the local level. Within this operational structure, the local fire chief has the primary operational responsibility for the on-site coordination of local disaster response. If the magnitude of an incident increases, then a regional coordination team will be formed in liaison with the operational coordination team at site. The regional coordination team is often situated in a regional office remote from the incident (e.g. a joint office of the regional emergency services). If a regional coordination team is formed, then the mayor of the municipality in which the incident is taking place takes the administrative lead. On municipality level, a policy team is formed to support the mayor.

Many more structures can be involved in ‘managing’ when the disaster incident transcends administrative borders e.g. a municipal, provincial or national border. When the potential magnitude of an incident leads to a serious threat to a large section of the community, environment, or property, emergency officers at provincial or national level are informed. If the effects of an incident transcend provincial borders (e.g. a toxic cloud after a nuclear incident) the Ministry of Internal Affairs may take the administrative lead. They will work together with coordination teams at national, provincial, regional and local level to manage and mitigate the disaster. For example, in an imminent flood, experts from the Water Board, the Ministry of Transport, Public Works, and Water Management will be involved in these coordinating teams. In case of major flood, a cooperation up to the international level may be established, e.g. during flooding from the river Rhine or Meuse, cooperation with Belgium and Germany was established (Rosenthal, Duin, Bezuyen, Vreeze-Verhoef, and Hart, 1998).

Actors and Responsibilities

Response and short-term recovery can be categorized into four different clusters, namely containment and control of the disaster and its effects, public order and traffic management, medical assistance, and taking care of the population. A cluster is made of several processes that are responsibility of mainly one of emergency response sectors, and may involve third parties when needed.

Containment and control of the disaster and its effects

Fire brigade units have an important role in the containment and control of a disaster and its effects. In the Netherlands, the fire brigade is usually organized at a municipal level and has equipment not only for fighting fire, but also for performing various measurements related to release of dangerous substances in the air, water or in the soil. The fire brigade is also responsible for alarming the citizens is case of emergency using the net of stationary sirens. Generally the fire brigade is obliged to maintain a fire brigade call centre, but the tendency of the last years is to maintain a common call centre (together with the police and GHOR). Usually, the fire brigade duty officer takes the lead in all small-scale emergencies (before the operational team is formed).

Several other organizations may also take a part in the containment and control, if the operational organizations (i.e. first responders) need support. For example, in case of flood (a major threat in the Netherlands) Directorate-General for Public Works and Water Management (Rijkwaterstaat, http://www.rws.nl/), the Dutch National Reserve (http://www.natres.nl/), the Royal Dutch Water Life Saving Association (KNDRD, http://www.rednet.nl), the Royal Netherlands Sea Rescue Institution (KNRM, http://www.knrm.nl) and Search and Rescue (SAR, http://www.werkenbijdemarine.nl/) can be involved. Some of these institutions, e.g. KNRM, SAR follow emergency scaling, which differs from the ones described in WRZO.

Public order and traffic management

In case of an emergency, the police are responsible for processes that are related to evacuation of citizens from affected areas, clear threatened areas, protect shelters and commando centers, controlling traffic, etc. In most cases the police are working under the authority of the mayor. In some special cases (e.g. criminal cases), the High Officer of Justice is taking the lead (together with the mayor and the regional police chief).

Medical assistance

The third large cluster comprises processes related to medical assistance. Key actors are the Ambulance Central Point (CPA), ambulances, hospitals, and Communal Health organization (responsible for general health issues such as prophylactic medical inspections, vaccinations, etc.). Compared to the fire brigade and police, the medical help is quite independently organized and does not directly depend on any local administrations. In case of emergency, however, a regional medical official is appointed who takes the lead within the medical help (similar to the regional officers in fire brigade and police structures). The hospitals are seen as trauma centers, where during disaster mobile medical teams (MMT) should be available 24 hours per day. One hospital is dedicated to the victims of disasters. The SIGMA teams of the Netherlands Red Cross (NRD, http://www.rodekruis.nl/) and special ambulance teams can be formed and included in the medical help operations. NRD is usually involved only in large disasters, requiring help and evacuation of many people, such as floods.

Taking care of the population

Besides the overall responsibility for disaster management (under the authority of the mayor), the municipal structures are responsible for processes related to taking care of the population such as informing citizens, accommodating non-injured people from affected areas, registering casualties, etc. Generally the municipalities have to take care of good preparation of response sectors as well as citizens. Therefore, the municipality has to prepare (and update every 4 years) the disaster management plan. The disaster management plan describes the most important types of disasters at the territory of the municipality and the way of dealing with a particular emergency. Responsibilities, tasks and all required medicaments, shelters, reserves of food and clothes, etc. are also part of the disaster management plan.

Information Needs

The information needed for emergency response is studied at different levels (locally, at provincial and national levels) and reported in many publications (mostly in Dutch). Generally, the information is grouped in two large clusters, dynamic (operational) and static (existing). Data collected during the disaster are denoted as dynamic data, while the information existing prior the disaster is named static information (Diehl, Neuvel, Zlatanova, and Scholten, 2006; Grothe, Landa, and Steenbruggen, 2008; Scholten, Fruijter, Dilo, and van Borkulo, 2008). Information needs, mainly reflecting requirements of the first and second cluster, are summarized in Table 1. Existing information is given for completeness, and is described in a more general level.

Operational information	Existing information
Incident: location, nature, scale, location of fire brigade Effects: affected area and its development in time, damaged objects, damaged infrastructure Consequences (estimates): threatened area (+time/period), escalation possibility Surroundings of the incident: sectormal (a diagram for first estimates of effected areas), gas plume Casualties: injured people, wounded, trapped, people & animals to feed ... Meteorological info: wind, precipitation, humidity … Accessibility: in- and out-routes, traffic direction, blocked roads Temporary centers: places for accommodating people (and animals),	Reference data: topographic maps, aerial photographs Managerial and administrative data: census data, administrative borders (municipality, region, province), risk objects (gas stations, storage places of dangerous goods, etc.), vulnerable objects (schools, nursing homes, prisons, etc.) Infrastructure: road network, utility networks (gas, water, electricity), parking places, dykes, etc. Buildings: high/low-rise material,

treatment centers, relief centers, morgues, … Decontamination: decontamination centers; vehicles, houses and infrastructure to decontaminate; people and animals to decontaminate Specific info depending on type of disaster, e.g. in case of: flood – velocity and water depth, flood pattern, ... incident with ships – ship type, numbers of people on board, owner, other ships in the surroundings, … aircraft incident – type of plane, function (cargo /military /civilian), number of people on board, type of fuel + volume, if a military plane: armed, …	number of floors, usage (residential, industrial), presence of dangerous materials, owners, cables and pipes, etc. Accessibility maps: for buildings, industrial terrains, etc. Water sources: fire hydrants, un-covered water, drilled water well, capacity, etc.

Table 1. Information needs, mainly reflecting requirements from the fire brigade and police.

Dynamic information is collected from processes of one cluster (i.e. from actors responsible for the cluster) and used inside this cluster and the other clusters as well. This information being, e.g. parameters of incident such as scale, development, which are updated regularly; number of victims considering different categories such as trapped, injured people, slightly wounded, death, missing, is also updated regularly; in case of detection of dangerous substances in the air, water or in the ground, special measurement teams are sent to collect samples – results of the measurements are reported to the commando and control centre and analyzed by a specialist. Some dynamic information has to be gathered from other organizations. For example, information about the actual water levels and the likelihood of a flood are to be obtained from Rijkswaterstaat. To support forecasting, numerous scenarios have been developed to give insight into possible flood consequences, and models have been developed to calculate the possibilities to evacuate people. These models are run and continuously adapted real-time when necessary (e.g. German/Netherlands system VIKING, http://www.programmaviking.nl).

As it can be seen from the above examples, the information used during disaster management is very wide, and of a very different nature. The model that is described in this paper is restricted to information collected by first emergency responders. It is part of a large framework that aims at offering information and services for the emergency response (Scholten et al., 2008). For the moment, the model is reflecting information collected by the fire brigade and police, but it is not covering information specific for some disaster (sub-)types, e.g. traffic incidents with ships, incidents with aircrafts. The model is to be extended later with this kind of specific information, as well as information collected by medical service and municipality.

MODELLING OPERATIONAL DATA

We are using Unified Modeling Language (UML, version 2.1) from Enterprise Architect (EA) software modeling tool for data modeling purposes. Oracle 10g is the DBMS system we use for storing the operational data. Modeling is done in three levels: conceptual level, logical level that is a translation of the previous to Oracle Spatial tables, and SQL scripts for creating types, tables, constraints, indices, and views in Oracle Spatial. Translation from a higher level to the next level is done partially automatic from inside EA, with additional modifications. The first two levels are built using class diagrams in EA. Presentation of the model in this paper is done in two steps. Section 3.1 presents an overview of the model, classes (without attributes) and their associations, using the conceptual model. Section 3.2 goes into more detail, adding attributes, and constraints. It uses the logical level, Oracle tables, and parts from SQL scripts for explaining constraints.

Overview of the Data Model

When a disaster incident happens, departments from emergency response sector get involved in order to manage the incident. Usually complaints are coming to the commando center, which report about the incident. Based on the type of the incident, several processes are activated, each process being responsibility of one or more departments, dependent on the scale of the incident. Several people get involved in these processes having specific roles. Also, several teams are formed in order to perform specific tasks. When the incident involves release of dangerous substances, a template named sectormal is used to sketch the zone affected by gas distribution. Several measurement teams are formed and sent to specific locations to perform measurements, from which the movement of gas plume is derived. An incident usually comes with casualties in people or animals, as well as damage to infrastructure.

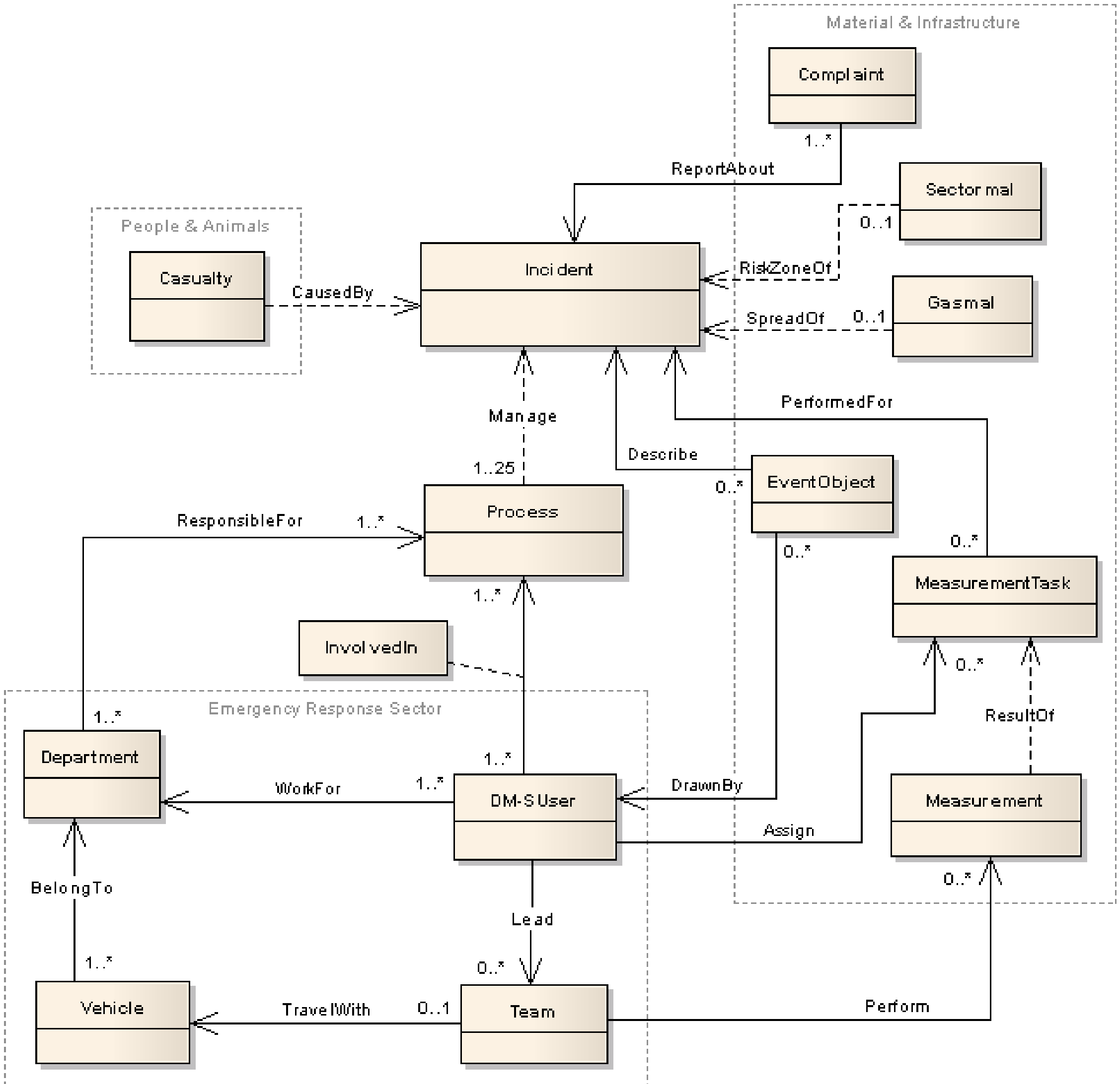

Figure 1. Conceptual level of operational data for disaster management in NL: classes (in boxes) and their relations (arrows together with a box in case it has attributes). Gray dotted line boxes show categories of information.

The conceptual model shown in Figure 1 captures classes of information and their associations. Multiplicity for associations is shown only when it is different from 1, thus a missing label indicates a multiplicity equal to 1. Dashed lines show dependency, which in our case means a source class which existence depends on the target class. Classes can be grouped into three categories based on the information they keep: information about people and animals, information about materials and infrastructure, and information about the emergency response sector. Gray dotted line boxes show these three categories. The last category is an example of the combination between existing and dynamic information. Departments, vehicles, and people from the emergency response sector belong to existing information. For each of these three classes we keep minimal information, basically a code to make the link between the operational data and the existing data. Dynamic information in this category is the involvement in processes managing an incident.

The rest of this section goes through all classes and associations of the conceptual model, giving a short explanation for each one. Class `Incident` contains information about a disaster incident, e.g. its location, its scale, disaster type.[1] Class `Process` contains information about processes activated to manage an incident, and association

[1] The complete list of attributes for each class is given in the logical level, the next section.

`Manage` keeps the relation with the incident they handle. `Casualty` contains statistics (counts) on victims of the incident, being those missing, injured, needing shelter, etc. Class `Complaint` contains information reported about an incident; several complaints can be done for an incident, as shown from the cardinality of `ReportAbout` association. These complaints come from citizens, often reporting via telephone about visible smoke, or strange smell, or breathing problems. The centralist has to register the location (address) and type of the complaint. Class `Sectormal` contains information about zones that will possibly be affected by an incident involving dangerous substances. This is done by marking circle sectors in a fixed template. `Gasmal` contains information about the gas plume for the incident. Class `EventObject` contains drawings done by system users to locate different events happening in the field, e.g. a gas leak, blocked road, damaged building, etc. Different measurements are performed for incidents that involve dangerous substances. A measurement task is designed by a specialist of dangerous substances (AGS), and sent to a team that performs the measurement according to task specifications. Class `MeasurementTask` keeps information about such task, the association `PerformedFor` keeps track of the incident for which a task was designed, and association `Assign` keeps track of which (AGS) user assigned what task. Class `Measurement` contains results of the measurements, and the association `Perform` records which team performed what measurement. A gas plume is derived from calculations on several measurements.

Class `Department` contains information about a department unit. A department is responsible for several processes started for the same incident or processes of different incidents. When the scale of an incident is big, several department units take the responsibility over the process. Association `ResponsibleFor` keeps track of responsible departments for each process. A department owns one or more vehicles, e.g. a fire brigade owns trucks or boats. Class `Vehicle` keeps information about all vehicles, and `BelongTo` takes care of the ownership. Class `DM-SUser` contains information about system users, i.e. people from the emergency response that are users of the DM system. A system user is involved in different processes of one or different incidents at different times. The association `InvolvedIn` contains the duration of such involvements. Association `DrawnBy` keeps track of objects drawn by each user. During response to a disaster incident, several teams are created with people from the emergency response sectors, and volunteers. Class `Team` keeps information about teams, e.g. number of its members, position of the team. We assume a team is created on spot, and has a one-time existence. Not all persons from a team are necessarily users of DM system, but there is one person who has access to the system. The association `Lead` indicates the team member that is a DM system user. A team uses a vehicle to travel to the place of incident, which is captured by `TravelWith` association.

Data Model in More Detail

In this section we explain the data model in more detail, providing attributes and constraints. At the same time we are passing to a lower level of modeling, DBMS tables. To keep the coherence between sections, we provide shortly the (result of translation as) correspondence with the conceptual model. Before going to the description of tables with their attributes and constraints, we need to introduce domain values for those attributes that are not of standard types provided by a (spatial) DBMS. There are agreements coming from the Dutch law, or others decided by emergency sector departments that are related to DM information. These are, e.g. 19 disaster types, with additional subtypes for some of them, 25 DM processes, 5 GRIP levels. In addition, there are attributes having a temporal or spatiotemporal nature. We have organized the data model into three components: static data reflecting DM agreements, new data types, and the tables containing the operational data (see Figure 2). This separation into components also reflects the different treatment of these components in Oracle Spatial. For each item in the `DMagreements` component we have created a look-up table containing all the possible values, e.g. `DMProcess` contains 25 records with information for each process. Later in this section we provide an example that shows how the look-up tables are used. The second component, `SpatioTemporalTypes`, contains the new data types for temporal and spatiotemporal data, and is explained in detail in the next section. In this section it is simply used to show that an attribute is of this temporal or spatiotemporal type. The third component, `OperationalDTables`, contains the tables for operational data, and is explained in the coming paragraphs.

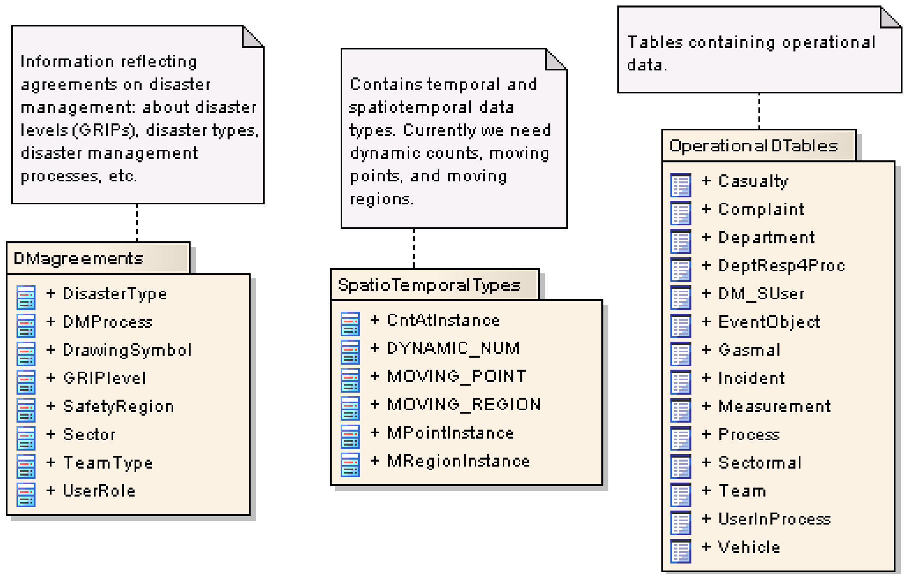

Figure 2. Organization of data model into components: look-up tables, new data types, and operational data tables.

Figure 3 shows the UML diagram for the logical level of our data model. For each class in the conceptual level there is a corresponding table in the logical level. There is one exception: we have merged classes `MeasurementTask` and `Measurement` into one table named `Measurement` to be able to express constraints using attributes of both classes. A new table is created for each many-to-many association: table `DeptResp4Proc` resolves the `ResponsibleFor` association, table `UserInProcess` resolves `InvolvedIn` association. All other associations are resolved by primary key – foreign key relations. The label `PK` in front of a column name indicates that it is (part of) a primary key, label `FK` indicates that it is a foreign key, label `pfK` indicates that it is (part of) a primary key that is also a foreign key, and `*` indicates that the column cannot be empty. Attributes are shown together with their type; `MDSYS.SDO_GEOMETRY` is the spatial type of Oracle Spatial. For the sake of space, we do not explain each attribute of every table. We describe the main tables and attributes, and in more detail attributes that are part of constraints. We hope that naming of an attribute and the table containing it, allow a sufficient understanding in view of the whole model.

Table `Incident` contains: an identifier (ID); location of the incident; the area affected by an incident as a dynamic region (spatiotemporal type); an estimation for the area threatened by an incident, a dynamic region as well; scale of the incident; to what disaster type it is classified, which may change in time; its GRIP level, possibly changing in time; the area to be cleared around the incident and the fenced area; the escalation risk described in words. Table `Process` contains: the identifier of a process as combination of the ID of the incident for which it is activated and the type of process, `process_type`; the duration of a process stored as start and end time of the process. The attribute `process_type` takes values from a look-up table `Process_Type` containing information for all processes. Creation of the table:

```
CREATE TABLE Process_Type (
        code NUMBER(2) CONSTRAINT PK_Process_Type PRIMARY KEY,
        value_EN VARCHAR2(70),
        value_NL VARCHAR2(70)
);
```

is done before the DM system is made available to emergency response, and the information is filled for all 25 processes. Values of `process_type` attribute are constrained to codes available in `Process_Type` table, by its declaration inside `Process` table:

```
process_type NUMBER(2) CONSTRAINT REF_Process REFERENCES Process_Type
```

All other items in `DMagreements` package (see Figure 2) are processed and used by operational tables in the same way.

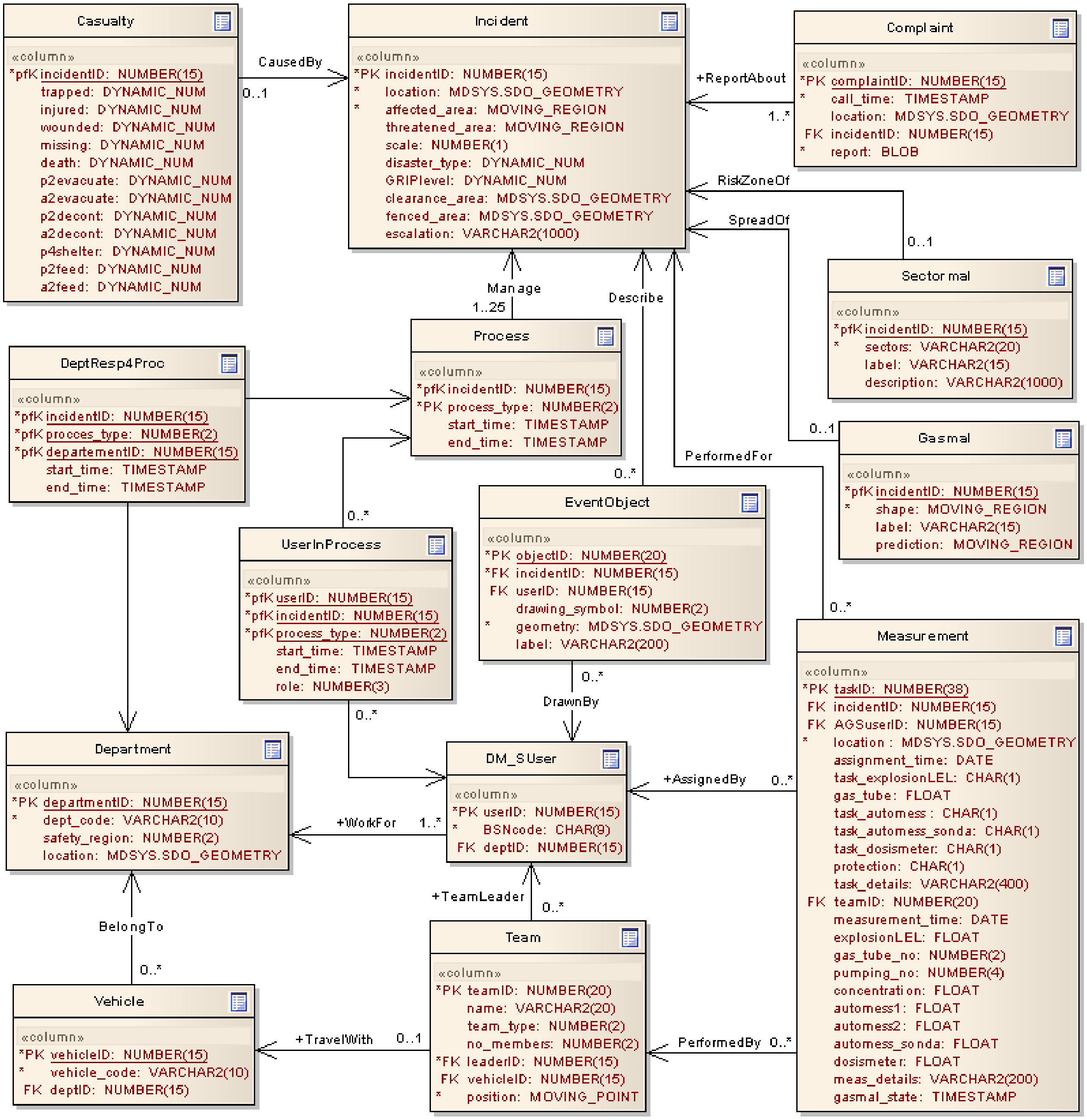

Figure 3. Operational data tables: PK indicates primary key; FK indicates foreign key; pfK indicates a column that is part of a primary key, as well as a foreign key; arrows show foreign key – primary key relation.

Table `Casualty` contains different counts for people or animals affected by an incident, e.g. number of people trapped during an incident, number of injured people, number of people or animals to decontaminate, e.g. during an incident involving toxic substances, number of people needing shelter, etc. Table `Gasmal` contains: ID of the incident that created the plume, shape of the plume as a dynamic region, a description label, and plume prediction for a fixed period, also as a dynamic region. Attributes in `Measurement` table reflect specifics of measurements performed by fire brigades in NL, which follow a standard form. The table contains an identifier; reference to the incident for which it is performed, reference to AGS that designed the task; reference to the team that performed the measurements; location where measurements are (to be) performed; the rest are mostly corresponding attributes for task and result of measurements. For example, explosion state LEL is a measurement from the form: attribute `task_exposionLEL` contains values yes/no deciding if this measurement should be performed, and attribute `explosionLEL` contains the measurement value in case the task decided that this measurement should be performed. This is checked through constraints written in the declaration of `Measurement` table:

```
task_explosionLEL CHAR(1) CONSTRAINT CHK_explosion_task
CHECK (task_explosionLEL in ('Y', 'N')),
…
CONSTRAINT CHK_explosion CHECK (task_explosionLEL <> 'Y' OR explosionLEL IS NOT NULL)
```

The first declares `task_exposionLEL` attribute as having only values '`Y`' (yes) and '`N`' (no); the second assures that if `task_explosionLEL` is set to yes, then `explosionLEL` is not empty. The compliance of the other performed measurements with those defined form the task is assured by similar constrains in the corresponding attributes.

Tables `Department`, `DM_SUser`, and `Vehicle` contain an identifier for our DM system, and a code (`dept_code`, `BSNcode`, `vehicle_code`, respectively) that relates to unique values in other databases in national or department level from where complete information can be found for each class. `Department` contains further the safety region in which a department falls, and the spatial location of the department. `DM_SUser`, and `Vehicle` contain in addition the reference to the department to which they belong. Table `Team` contains: an identifier, name, what kind of team it is, number of members, ID for the team member that is a system user, and of the vehicle they use, and team position that is dynamic.

Parts of the data model are also indices, including spatial indices, and new data types. We do not discuss here the indices created in the tables, with the exception of spatial indices found in the next section. New types are introduced in the next section, which goes further on handling data using these new types.

TEMPORAL AND SPATIOTEMPORAL DATA

This section provides for creation of new data types in Oracle, and their use in tables in Section 4.1, reading and writing spatiotemporal data into tables in Section 4.2.

Declaration and Use of New Data Types

Three new data types are needed for the data model provided in the previous section: `DYNAMIC_NUM` for dynamics counts, e.g. number of injured people; `MOVING_POINT` for dynamic points, e.g. position of a team in the field; `MOVING_REGION` for dynamic regions, e.g. the gas plume. A dynamic count is stored as a sequence of pairs (cnt_i, t_i), where cnt_i is the count value at instance t_i. For any time *t*, the value for count can be calculated from linear interpolation between two consecutive counts cnt_i and cnt_{i+1}, such that $t \in [t_i, t_{i+1}]$. Similarly, `MOVING_POINT` is stored as a sequence of pairs, point location and time, and `MOVING_REGION` as a sequence of pairs, polygon shape and time. Different interpolation techniques can be used for calculating point position and polygon shape for any moment of time, see e.g. (Meratnia 2005; Meratnia and de By, 2003) for moving point interpolation. This is a full subject by itself, and is out of the scope of this paper.

We use nested tables in Oracle to store sequences for dynamic types. For example, to create `MOVING_POINT` type, first an object type is created containing a time instance and point location, which is then used to build the sequence as a table of such objects:

```
CREATE OR REPLACE TYPE MPointInst AS OBJECT (
        meas_time TIMESTAMP,
        point_geo MDSYS.SDO_GEOMETRY
);
/
CREATE OR REPLACE TYPE MOVING_POINT AS TABLE OF MPointInst;
/
```

The other types, `MOVING_REGION` and `DYNAMIC_NUM` are created in a similar way.

The new data types are used in tables containing attributes of a dynamic nature. For example, `Team` table has attribute team position that is a moving point. Below is the statement for creating the table containing an attribute of type moving point:

```
CREATE TABLE Team (
        teamID NUMBER(20) CONSTRAINT PK_Team PRIMARY KEY,
        name VARCHAR2(20),
        team_type NUMBER(2) REFERENCES TeamType,
```

```
        no_members NUMBER(2),
        leaderID NUMBER(15) REFERENCES DM_SUser,
        vehicleID NUMBER(15) REFERENCES Vehicle,
        position MOVING_POINT
)
NESTED TABLE position STORE AS TeamPosition;
```

Other tables, like `Incident`, `Gasmal`, `Casualty`, which contain dynamic attributes have similar declaration.

Working with Spatiotemporal Data

This section starts with writing data into spatiotemporal attributes, continues with reading data from tables containing spatiotemporal data, and concludes with examples of use of spatiotemporal data in analysis. Different examples of SQL statements are provided for reading and writing data into tables.

A spatiotemporal attribute can be filled by adding values (point location or polygon shape) for each time instant one by one. First an empty table should be created for the spatiotemporal attribute, and then time-geometry pairs can be added one by one. For example, assume a new team is created, and its data is entered into `Team` table, together with an empty instance for the spatiotemporal attribute `position`:

```
INSERT INTO Team(teamID, name, team_type, no_members, position)
        VALUES(12,'Whisky 349', 2, 4, MOVING_POINT());
```

Once the empty nested table is created, records can be added to it:

```
INSERT INTO TABLE(
        SELECT t.position
        FROM Team t
        WHERE TeamID = 12)
VALUES(sysdate, sdo_geometry(2001, 90112,
        sdo_point_type(86875.2, 447457.9, null), null, null));
```

The statement above adds one record, time-position pair, where time is the current system time, and position is point type geometry of Oracle Spatial. A block of values can be written at once in a spatiotemporal attribute. We have GPS logs giving position of a team in the field, and have put the data in a temporary table `GPStrack`, which contains:

```
desc GPStrack;
Name                  Null?          Type
------------------    ------------   -----------------------
USERID                               NUMBER(11)
TIME_                                CHAR(19)
GEOMETRY                             MDSYS.SDO_GEOMETRY
```

To insert the data from this flat table into the `Team` table in our model, e.g. for team with ID = 16, the following statement is executed:

```
INSERT INTO Team(teamID, position) VALUES(16,
        CAST(
        MULTISET(SELECT MPointInst(time_, geometry)
                FROM GPSTrack
                WHERE userid = 16
                ) AS MOVING_POINT
        )
);
```

Each individual record of a spatiotemporal attribute can be accessed by performing an un-nesting of the table. For example, to view the trajectory of team 16 that was entered above, we execute the statement:

```
SELECT tp.meas_time, tp.point_geo
FROM THE(SELECT position
        FROM Team
        WHERE teamID = 16) tp;
```

For the purpose of visualization, spatiotemporal data is converted to simply spatial data. We create views that perform un-nesting of tables, in order to turn the spatiotemporal types to spatial types. For example, to view data from `Team` table, `TeamTracking` view is created:

```
CREATE                    VIEW                    TeamTracking                    AS
SELECT   t.teamID,   t.team_type,   t.name,   t.no_members,   t.vehicleID,   p.*
FROM Team t, TABLE(t.position) p;
```

which structure is

```
desc                                                                    TeamTracking;
Name                    Null?          Type
-------------------     -----------    -------------------
TEAMID                  NOT NULL       NUMBER(20)
TEAM_TYPE                              NUMBER(2)
NAME                                   VARCHAR2(20)
NO_MEMBERS                             NUMBER(2)
VEHICLEID                              NUMBER(15)
MEAS_TIME                              TIMESTAMP(0)
POINT_GEO                              SDO_GEOMETRY()
```

To visualize data from Oracle Spatial, metadata should be filled for spatial attributes. For the view created above we add metadata information for the spatial attribute, specifying the extent and the reference system (for the whole Netherlands):

```
INSERT INTO USER_SDO_GEOM_METADATA VALUES
       'TEAMTRACKING',
       'POINT_GEO',
       SDO_DIM_ARRAY(
              SDO_DIM_ELEMENT ('X', 0, 220000, 0.05),
              SDO_DIM_ELEMENT ('Y', 250000, 630000, 0.05)),
       90112);
```

Data of `TeamTracking` view can be visualized from FME software (or any software visualizing Oracle Spatial data), and can also be accessed from a web service.

The spatiotemporal data is to be used in analysis, in order to help decision-making during disaster management. Analysis usually requires combination of existing data with the operational data, which in turn requires combination of spatial functionality (on static data) with spatiotemporal functionality. For example, consider a fire-brigade team that takes the order to go to an incident. A shortest path algorithm is run on the road network (existing data), considering starting location of the team, and the location of the incident. While the team travels towards the incident location, the time of arrival at the incident is continuously calculated based on the current position of the team – a spatiotemporal functionality using spatiotemporal data (team position) and static data (calculated path from the road network). Parts of the road network can get blocked as a consequence of the incident, or other reasons as well. The best route (i.e. shortest path) is to be re-calculated on the available network. This example, and many others coming from the services required by the emergency response, require spatiotemporal functionality built over the spatiotemporal data types presented in this section.

CONCLUSIONS AND FUTURE WORK

This paper provided a model for data collected during emergency response. The model reflects the organization of emergency response in the Netherlands, based on law agreements. It captures the management of disaster through processes; the involvement of response sectors in these processes; the consequences of an incident in people and animals; consequences in infrastructure. The data model was first built in a conceptual level, and finally implemented in Oracle Spatial. As most of the operational data is dynamic, temporal and spatiotemporal types were created to store such data as nested tables. We showed how to write and read the spatiotemporal information using Oracle functionality for nested tables.

This model was created based on information requirement coming mainly from the fire brigade, and partially from police. These requirements were not exhaustively reflected in the data model. Furthermore, there are two other sectors in the emergency response: medical service and municipality, which information needs have to be studied, and reflected in the data model. The model will be extended in order to cover all the information needs. In addition to this, we are working on providing functionality for spatiotemporal data. The two directions are independent, and we are working in parallel in the extension of the data model (possibly with modifications) and functionality for spatiotemporal data.

ACKNOWLEDGMENTS

This work is part of the project Geospatial Data Infrastructure for Disaster Management (GDI4DM) funded by RGI program of the Dutch government. The authors want to thank partners from this project for several discussions about the data model, as well as colleagues from GIS-technology section of TU Delft for their comments and suggestions.

REFERENCES

1. David, E., Evaux, L. and Masson, A. (2006) Flood propagation in urban areas: new involvement of modeling approach and perspectives, *Proc. 7th International Conference of Hydroinformatics*, 26-xx, Research Publishing, Singapore.
2. Diehl, S., Neuvel, J., Zlatanova, S. and Scholten, H. (2006) Investigation of user requirements in the emergency response sector: the Dutch case, *Second Symposium on Gi4DM*, 25-26 September, Goa, India, CD ROM, 6p.
3. Erlich, M. (1996) Feasibility of the supervision and steering on a storm drainage system through rainfall scenarios recognition, *Stochastic Hydraulics'96*, 395-402, Balkema, A.A., Rotterdam.
4. Grothe, M.J.M, Landa, H.C. and Steenbruggen, J.G.M. (2008) The value of Gi4Dm for transport and water management, in: Zlatanova and Li (Eds.) *Geospatial Information Technology for Emergency Response*, Taylor & Francic Group, London, 313-328.
5. Meratnia, N. (2005) Towards Database Support for Moving Object Databases. PhD thesis, Twente University, The Netherlands.
6. Meratnia, N. and de By, R. A. (2003) Trajectory Representation in Location-Based Services: Problems and Solution, in: Santucci, Klas et al. (Eds.) *Proceedings of Fourth International Conference on Web Information Systems Engineering Workshops (WISEW'03)*, 18-24.
7. van Oosterom, P. and Lemmen, C. (2006) The Core Cadastral Domain Model: A Tool for the Development of Distributed and Interoperable Cadastral Systems, in: Villikka (Ed.) *Proceedings of the International Conference on Enhancing Land Registration and Cadastre for Economic Growth*, 1-12.
8. Quak, W., van Oosterom, P., de Vries, M., Bruns, B., Bakker, N., Reuvers, M. and Janssen, P. (2005) TOP10NL in GML3 (in Dutch), in: *Geo-Info*, Volume 2, 11, 504-510.
9. Rosenthal, U., Duin, M., v. Bezuyen, M., Vreeze-Verhoef, M. L. and Hart, P. (1998) Flood management in the Netherlands, in Rosenthal et al (Eds.) *Flood response and crisis management in Western Europe: a comparative analysis*, 103-152, Springer, Berlin.
10. Scholten, H., Fruijter, S., Dilo, A., and van Borkulo, E. (2008) Spatial Data Infrastructure for emergency response in Netherlands, in *Remote Sensing and GIS technology for monitoring and prediction of disaster*, Nayak, Zlatanova (Eds.), 177-195, Springer-Verlag.
11. Snoeren, G. (2006) Rampbestrijdingsprocessen: Actoren, werkwijze, data. GDI4DM technical report, 115 p.
12. Xu, W. and Zlatanova, S. (2007) Ontologies for Disaster Management Response, in *Geomatics Solutions for Disaster Management*, Li, Zlatanova, and Fabbri (Eds.), Heidelberg, 185-200.
13. de Vries, M.E., Tijssen, T.P.M., Stoter, J.E., Quak, C.W. and van Oosterom, P.J.M. (2002) The second GML prototype of the new TOP10vector object model, GISt Report No. 12, Delft, 2002, 148 p. available at http://www.gdmc.nl/publications