IOP Conf. Series: Journal of Physics: Conf. Series **801** (2017) 012037 doi:10.1088/1742-6596/801/1/012037

A Conceptual Data Model for Flood Based on Cellular Automata Using Moving Object Data Model

R S Rachmatullah[1], F N Azizah[2]

School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Jln. Ganesha no. 10 Bandung, Indonesia

[1] 13512036@std.stei.itb.ac.id, [2] fazat@stei.itb.ac.id

Abstract. Flood is considered as the costliest natural disaster in Indonesia due to its frequent occurrences as well as the extensive damage that it causes. Several studies provide different flood prediction models based on various hydrological factors. A lot of these models use grid-to-grid approach, making them suitable to be modelled as cellular automata. This paper presents a conceptual data model for flood based on cellular automata model using spatio-temporal data model, especially the moving object data model, as the modelling approach. The conceptual data model serves as the model of data structures within an environment for flood prediction simulation. We describe two conceptual data models as the alternatives to model the data structures of flood model. We create the data model based on the study to the factors that constitute the flood models. The first conceptual data model alternative focuses on the cell/grid as the main entity type. The changes of the states of the cells are stored as moving integer. The second alternative emphasizes on flood as the main entity type. The changes of the flood area are stored as moving region. Both alternatives introduce some advantages and disadvantages and the choice rely on the purpose of the use of the data model. We present a proposal of the architecture of a flood prediction system using cellular automata as the modelling approach. As the continuation of this work, further design and implementation details must be provided.

1. Introduction

Flood is one of natural disasters that frequently occur in Indonesia. According to the data of Badan Nasional Penanggulangan Bencana Indonesia (Indonesia's National Agency for Disaster Management), up to the month October, there have been 639 occurrences of flood throughout Indonesia in 2016 in which 139 people died or missing, more than 2 million people were displaced temporally or permanently from their houses, as well as hundreds of millions of rupiahs of material loss [1]. Due to the frequent occurrences and extensive damage that it causes, flood is considered as the costliest natural disaster. In the attempt to reduce the loss, several initiatives have been taken by the Indonesian government, such as constructions of reservoirs, river dredging, and the use of flood prediction systems to determine the location and time of the incidents.

Several studies on flood prediction model use various hydrological cycle factors depending on the focus of studies. A lot of these models use grid-to-grid (G2G) approach to determine the spread of flood. A flood is modeled as the dispersal of water from one grid to its neighbors per unit of time [2]. This characteristic makes cellular automata a suitable approach to model it.

In this paper, we present a study on several existing flood prediction models to determine the important factors that make up a flood, as well as the formulas used to calculate the spreading of flood.

We especially examine how the cellular automata concept is used to model the flood. We incorporate the result of the study to create a conceptual data model using spatio-temporal approach, especially the moving object data model [3]. Moving object data model is chosen for its maturity as a spatio-temporal data modelling concept.

The conceptual data model serves as the model of the data structures within an environment for flood prediction simulation using cellular automata and spatio-temporal data modelling as the main approach. Eventually, our final goal is not only for simulating flood, but also providing a spatio-temporal model based environment for simulating problems that can be modeled as cellular automata, such as: fire (e.g. [4]). Flood, in this case, is an interesting case because it has a complex nature in term of different factors that we must deal with and complex formula for calculating the spreading of flood.

The paper is organized as the following: in the next chapter, we present some theoretical background regarding flood prediction models, cellular automata, as well as moving object data model. Subsequently, we discuss two alternatives of the conceptual data models based on the result of the study on the factors involved in the formation and spreading of flood. We present the impact of each alternative on the algorithms used for simulating the cellular automata as well as its subsequent implementation in the simulation environment.

2. Theoretical Background

2.1. Basic Equations for Flood Models

Flood is rainfall excess (surface runoff) that cannot be covered neither by infiltration nor by drainage system [2]. The flood models are usually created based on the hydrological cycle models that are used to model fluid and water circulation. An important key component of the model is the water balance equation which is calculated based on the Land Surface Model (LSM). The LSM is used to calculate the water balance in soil by using precipitation, infiltration, and evapotranspiration factors [5].

The water balance equation calculates the amount of water in soil so that the input and the output is balanced [2][5]. The input is precipitation, i.e. the total of the volume of water or rainfall per unit of time. The output is divided into three parts: infiltration, evapotranspiration, and runoff. Infiltration components are calculated based on the soil types and land use. Meanwhile, the evapotranspiration factor is divided into two components, namely the evaporation and the transpiration. Evapotranspiration can be calculated using Penman-Monteith equation stated in [6].

2.2. Cellular Automata Approach for Flood Spreading

The result of the calculation on the surface runoff from water balance equation is used by the inundation (flood) model to determine the flow of water and to create puddles [2]. The flow of water is determined using flat-water concept. The flat-water concept assumes that the nature of water is always flat without considering the friction caused by gravitation or inertia and influx or wave. Because of flat water concept, the spreading of flood can be determined just by using topographic data from the Digital Elevation Model (DEM) [2][7]. The cellular automata model approach is used to determine the spread of flood by utilizing the properties of the water that flows from the high place to a low place.

In representing the flood model using the cellular automata model, grids or cells are divided into two types: the wet cells for flooded soil and water objects such as river; and the dry cells for soil that has not yet been flooded [7]. The starting point of calculation using the grid-to-grid approach is determined from the cell that has the highest height of cell. Height of a cell is composed by the height of water and the topography. That cell will be the central cell. The neighboring cells have to be ready to accommodate any excess water from the central cell.

2.3. Moving Object Data Model

Spatio-temporal data poses a challenge in database management system since handling spatio-temporal data is not a straightforward task due to the complexity of the data structures. It requires careful analysis in dimensions together with the representation and manipulation of the data involved. Moving object data model [8] assumes the aspect of time as an integral part of the spatial aspects of the object. This modelling approach captures the change of geometry and the movement of the object. Therefore, it is suitable for all queries required in spatio-temporal problems.

The moving object data model introduces several data types. For each data type, operations are defined. They can be divided into two groups [9]: non-temporal data types and temporal data types. The temporal data types provide the combination of the base or spatial types with the temporal aspect of the data. The new types are called the moving data types, for example: moving real (*mreal*), moving int (*mint*), moving point (*mpoint*), moving line (*mline*) and moving region (*mregion*). Another temporal data type is the *intime* data type which is used to define a single pair of (*instant*, value).

3. The Modelling Methodology

This work is aimed at providing the conceptual data model of flood models which considers cellular automata as the main flood modelling approach and moving object data model as the spatio-temporal modelling approach. As can be observed in the previous section, various factors must be considered in the modelling of flood and therefore, they must examine further in order to provide an appropriate approach on how to incorporate them in the conceptual data model. The cellular automata model determines how we store the flood event in which cell states change over time depending upon the calculation of the surface runoff on each cell.

To present our conceptual data model, we use Entity-Relationship (ER) model which uses ER diagram as the main tool to visualize the data model. We incorporate the use of moving object data types in the data model as the types for the attributes of entity types. Figure 1 shows the methodology that we use in this paper to define the conceptual data model of

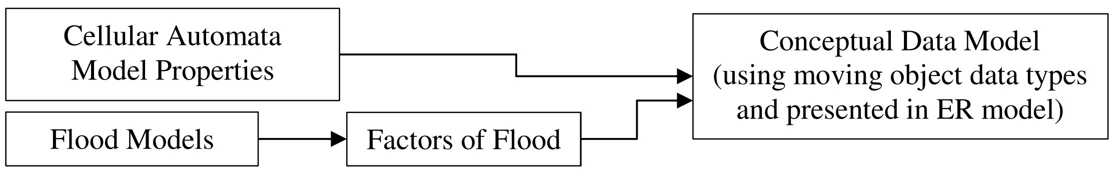

Figure 1. The Modelling methodology.

4. The Alternatives of Conceptual Data Model for Flood

4.1. Study on the Factors that Constitute Flood

Several studies on flood models use different formulas in the process and require different data. The more complex and complete the data, the more accurate the prediction results will be. As observed in water balance equation, rainfall or precipitation is the foremost data that must be provided. Evapotranspiration can be estimated using Penman-Monteith equation [6]. By contrast, Chen et al. [2] stated that evapotranspiration factor in urban areas can be ignored because it is not considered significant.

The calculation of infiltration can be done by using the VIC model of water balance in every layer ground [5]. The experiment conducted by Chen et al. [2] adds drainage capacity as a factor in infiltration capacity especially in urban areas. The experiments conducted by Prasetya et al. [7] uses the Horton equation to determine the capacity of water absorption per unit time. Standard absorption rate is determined as a constant, based on MODIS land cover classes. To determine the surface runoff and infiltration, additionally, a flood model requires the results of the morphology of satellite images or GIS (Geographic Information System) data to determine the soil type and MODIS land cover class or Land Use. Furthermore, the models require DEM data for ground topology. Subsequently, VIC model can be applied to calculate the distribution of water runoff only when all data are present. Other flood models may not require such requirement.

After examining the theories on the flood models, we conclude that some of the factors that determine the flood models are mandatory (it means that the data must be provided), while some others are optional (it means that they can be omitted when they are not present). The factors can also be viewed from the global or local perspectives. Global factors are shared and influence all process and cells in the cellular automata model. Local factors, on the contrary, influence and are used exclusively within a cell. Several factors can give a broad impact to the area of prediction, for example: daily mean temperature and wind speed. Therefore, they are designated as global factors.

4.2. The Proposed Conceptual Data Models

To model the factors that constitute a flood, we define several entity types, i.e. the Soil Reference, Soil Type, Land Use, Drainage, Topography and District. The entity Land Use defines the usage of a certain piece of land. It contains the attributes to define the type of land use (e.g.: forest, farm, field, residence) as well as the area (spatial type: region) covered by the land. The entity Soil Type defines a certain area of soil and what kind of soil type it belongs to. The attributes define for the Soil Type are the type name as well as the area covered (spatial type: region). The entity Soil Reference refers to the soil condition which can be imposed either to a Land Use or a Soil Type.

The existence of drainage systems is captured via the entity Drainage which contains the attributes the volume of the drainage, type, as well as the area covered by the drainage. The entity Topography is used to store the height of land area. The entity District covers a certain area, usually in the form of governmental district, in which we record the rain precipitation, the radiation flux, humidity, wind speed, aerodynamic resistance, mean saturated vapor pressure and daily mean temperature, all of which are of the type moving real as well as the air density. The attributes are modeled as moving real since the values changes over time. The ER diagram that covers the factors that constitute the flood models can be seen on Figure 2.

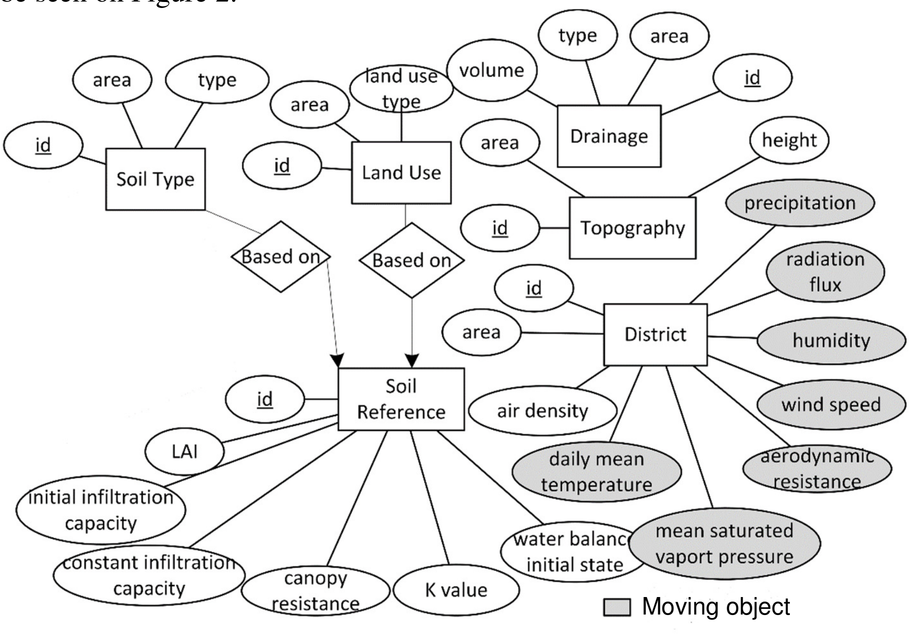

Figure 2. The ER diagram for the factors of flood.

To model the area affected by the flood, we define two alternatives of modelling approach:

1. focusing on the nature of cellular automata model, we store the cells and the changes of their states (as well as other attributes) over time;
2. focusing on the flood as an entity, we store the flood area and its changes over time.

Using the first approach, we store each cell as an entity. Thus, we have the Cell entity type. The state of a cell may be: (1) wet; (2) dry (not wet). A cell is in the state of wet when it is submerged by water. Since the state of each cell can change over time, it is stored as a moving object attribute (type:

moving integer). Other contributing factors are also measured: the water height and the water balance, both also changes over time. Thus, they are stored as moving object attributes (type: moving real). See Figure 3 (a) for the ER diagram of the model.

The second approach considers flood as an entity whose area changes over time. Thus, we define the Flood entity type. Therefore, we store the flood area as a moving object attribute (type: moving region). Other attributes to be stored for a flood are: the average of water height and water volume for the whole flood area per time. Both are also defined as moving object attributes (type: moving real). See Figure 3 (b) for the ER diagram of the model.

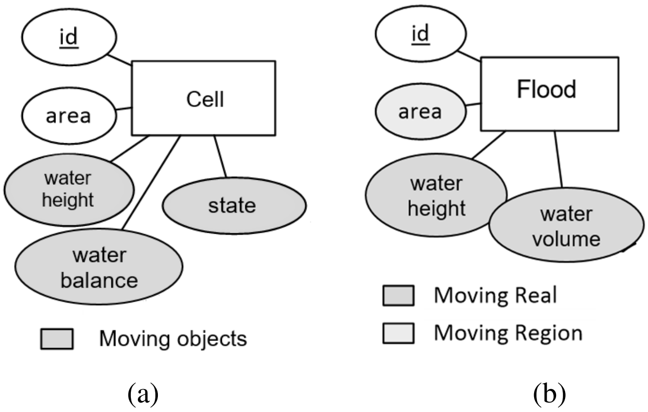

Figure 3. Two modelling approaches of flood as a cellular automata model: (a) 1st approach: focusing on the nature of cellular automata, cells are stored as a moving object entity; (b) 2nd approach: focusing on the flood as an entity, flood is stored as a moving object entity

How flood is stored influences the algorithms for simulating the flood using cellular automata approach. The first approach is obviously a natural way of storing the results of the cellular automata calculation. To calculate the state of each cell, we need to intersect the area of the cell with the areas of the other factors of flood: the soil type, the land use, the drainage, the topography and the region. Thus, we can get the value of each contributing factor and use the formulas to get the state of the cell.

The basic algorithm to do the flood simulation is as the following:

1. Intersect the area of each cell with the area of the entity Soil Type, Topography, Drainage and Land Use (the Soil Reference entity involves accordingly).
2. For each unit of time: do the calculation of the equation 1 (do first the calculation to get the evapotranspiration and infiltration, involving also the data of the entity District) and check if there is a runoff in each cell.
3. If there is a runoff, change the status of the cell to 'wet' and then give the water runoff to neighbor cells based on Prasetya et al. inundation algorithm [7] or Chen et al. algorithm [2].
4. Once the water runoff has been calculated and spread properly, keep the state of each cell for the unit of time in the attribute of state of a Cell, and then, re-do step 2.
5. Process 2-4 is repeated until the rainfall data are all processed or the simulation is terminated.

The second approach stores the changes of flood area over time. In each unit of time, after the state of each cell is calculated, all adjacent wet cells are combined to form the spatial object of flood. Therefore, we still require the steps to calculate the state of each cell for each unit of time, except that we are not required to store the state of each cell. Therefore, we need to adapt the 4th step of the algorithm into:

4. Once the water runoff has been calculated and spread properly, perform an aggregation operation on the area of wet cells to form an area of the flood per time and then store it as a part of the moving region of Flood. Subsequently, re-do step 2.

An important note to take is that calculating the state of cells per unit of time require a great amount of work. Suppose we have a n x n grid on the cellular automata model, then we must do n x n cell processing or in term of algorithm complexity, we will have a quadratic time complexity of $O(n^2)$. In this paper, we will not discuss how to deal with the challenge to create an efficient algorithm to reduce the time complexity required to calculate the cell states. Nevertheless, this challenge must be kept in mind when discussing the approach to store the data.

The first data modelling approach (model shown in Figure 3 (a)) provides the direct correspondence between the cells in the cellular automata and the cells in the data store. No further processing is required once the state of a cell per unit of time is determined: the resulting calculations of state, water height and water balance of each cell are directly stored in the appropriate attributes. As the calculation of the state of each cell requires great effort, directly store them will reduce the processing time. Suppose we have a n x n grids, then we will need a space to store n^2 cell entities, or the space complexity is $O(n^2)$.

Using the second data modeling approach (model shown in Figure 3 (b)), however, after we calculate the state of each cell, we need extra time and other resources to aggregate the wet cell area to form the area of a flood per unit of time. If we have n wet cells, the time complexity to aggregate the area will be $O(n)$. As mentioned earlier, calculating the state of the cells per unit of time can already be considered as a horrendous task. Adding the task to aggregate them into a flood area will not give a better picture. Nevertheless, in term of space requirement, for each flood event, we need only to store 1 (one) entity, i.e. the flood itself. Therefore, we can say that the space complexity is constant. The changes of the flood area are stored as a moving area attribute, i.e. for each flood, it is the aggregation of the wet cells. This will significantly reduce the amount of space required to store the data in comparison to storing the changes of all cells.

The second approach may also be more advantageous for further exploration and research. If we want to do further analysis on the flood data or do some statistics, especially the ones involving the area of the flood, the second approach may be more suitable than the first approach. Suppose we want to get the information about the coverage of a flood (at all times or at a certain time slice). Using the first approach, we will need to aggregate all the wet cells into the flood area. Using the second approach, the flood area has been provided in the storage.

The first approach is strictly tight to the grid-to-grid/cellular automata model. Thus, if we want to do simulation using different flood models that do not involve the grid-to-grid approach, we will find difficulties to adapt to the data structure. As for the second approach, since we store directly the flood, depending on how the flood model works, the problem to adapt to the structure will not be worse than the problem it has with the cellular automata approach. It will work even better when the flood model gives the flood area as its direct result.

Table 1 shows the summary of the comparison between the two approaches.

Which one of the two approaches is better? The answer, unfortunately, relies heavily on the purpose of the use of the conceptual data model. If the data structure is required in an environment in which simulation is purposefully done using the cellular automata model, then the first approach is favorable. The second approach, as discussed, actually provides a more general approach for more flood modelling and analysis.

5. The Proposed Architecture of The Flood Prediction System

As earlier described, the development of the conceptual data model of flood is aimed at providing the structure for the data associated with flood models and providing an environment on which experts can simulate flood events using cellular automata as the flood modelling approach. Figure 4 shows a proposal of the architecture for the simulation environment. The conceptual data model serves the blue print of the data stored in a moving object database management system (MO-DBMS). We reserve that no mature moving object database management system is ready to be used. Therefore, we propose the use of current existing database technology, equipped with the moving object data types and operations. On the application level, we have the flood prediction tool which consists of two important

IOP Conf. Series: Journal of Physics: Conf. Series **801** (2017) 012037 doi:10.1088/1742-6596/801/1/012037

modules: the Cellular Automata Module and the Visualization Module. During a simulation, the Cellular Automata Modules calculates the model based on the stored data and store the results back to the database. The Visualization Module visualizes the result of the simulation process.

Table 1. Comparisons between the conceptual data model alternatives.

	First Approach	**Second Approach**
Description	Focusing on the storage of cells, resulting in Cell as a moving object entity	Focusing on flood event, resulting in Flood as a moving object entity
Time complexity for cell calculation per unit of time	Cell processing: $O(n^2)$ n x n is the number of cells	Cell processing: $O(n^2)$ Aggregation of cells: $O(m)$ n x n is the number of cells m is the number of wet cells
Space complexity for one flood event simulation	$O(n^2)$ n x n is the number of cells	$O(1)$
Further information or statistics involving the flood area	Difficult, always need to aggregate the area of the wet cells	Easier, take the information from the moving area attribute of flood
Possibility to use other approaches for flood modelling	Difficult, since the model is tightly related to the cellular automata model	Easier to adapt since the model stores directly the flood data

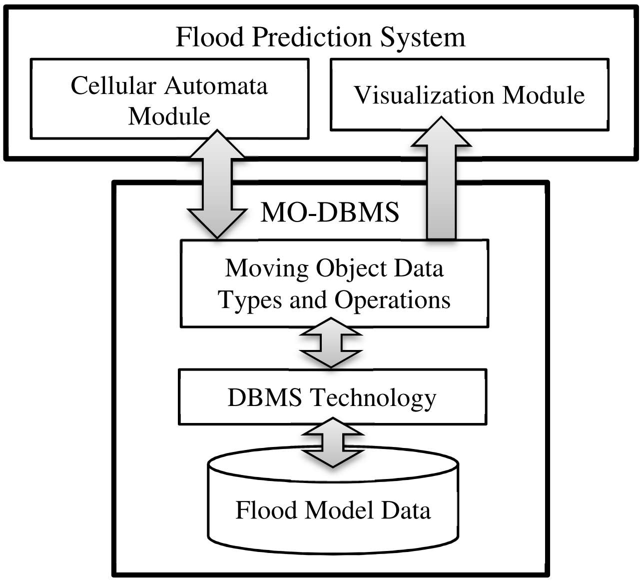

Figure 4. The proposed architecture of the flood prediction system

Since the system focuses on the use of cellular automata as the modeling approach, it is natural to use the first conceptual data model alternative (see Figure 3 (a)) as the appropriate data modelling approach. We will examine the modelling choices and the simulation architecture environment further as the continuation of this work.

6. Conclusions and Future Works

Flood simulation system is important in supporting the efforts of reducing the effect of flood as the costliest natural disaster in Indonesia. In this paper, we describe two conceptual data models as the alternatives to model the data structures of flood model using cellular automata as the main approach. We utilize the moving object data model approach to model the changes in the simulation environment. We create the data model based on the study to the factors that constitute the flood models. Several mandatory and optional factors of flood models are defined as the result of the study and based on that, a conceptual data model for the factors is defined. This data model is a common part for both conceptual data model alternatives.

The first conceptual data model alternative focuses on the cell/grid as the main entity type. In particular, it stores the changes of the states of the cells as moving integer. This approach produces a one-on-one correspondence between the cells in the cellular automata approach and the cells in the data storage, making it the best option for simulating the flood in a cellular automata environment.

The second alternative emphasizes on flood as the main entity type. It stores the changes of the flood area as moving region. In a cellular automata based simulation environment, this approach introduces extra effort to aggregate the flood area based on the wet cell area which must be done after the calculation of the states of each cell. On the other hand, this approach is probably a better choice for a more general purpose of flood modelling and analysis.

The conceptual data model serves as a basis for the development flood prediction system which involves simulating the flood models. We present a proposal of the architecture of a flood prediction system in section 5. Our next plan is to provide details in the implementation of the flood prediction system. We still need to provide the logical and physical design of the data as well as choosing and developing the appropriate database technology for storing the moving object data. We need to implement the cellular automata algorithm based on the chosen data design as well as providing a mechanism to visualize the result. Furthermore, we are still required to test the system to evaluate the accuracy of the model and its implementation.

References

[1] Badan Nasional Penanggulangan Bencana 2016 *Data dan Informasi Bencana Indonesia* http://dibi.bnpb.go.id

[2] Chen J, Hill A A and Urbano L D 2009 A GIS-based Model for Urban Flood Inundation *J. Hydrology* **373** (1–2) p 184–192

[3] Erwig M, Güting R H, Schneider M and Vazirgiannis M 1999 Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases *GeoInformatica* **3** (3) p 269–296

[4] Ohgai A, Gohnai Y and Watanabe K 2007 Cellular Automata Modeling of Fire Spread in Built-Up Areas—A Tool to Aid Community-Based Planning for Disaster Mitigation *Computers Environment and Urban Systems* **31** (4) p 441-460

[5] Liang X, Lettenmaier D P, Wood E F and Burges S J 1994 A Simple Hydrologically Based Model of Land Surface Water and Energy Fluxes for General Circulation Models *American Geophysical Union* **99** p 14415–14428

[6] Zotarelli L, Dukes M D, Romero C C, Migliaccio K W and Morgan K T 2010 *Step by Step Calculation of the Penman-Monteith Evapotranspiration (FAO-56 Method)* (Florida: The The Institute of Food and Agricultural Sciences)

[7] Prasetya R B, Fariza A, Hasim J A N and Basuki 2015 A River Flood Spreading Prediction System Using Cellular Automata (Case Study Bengawan Solo River) *Proc. Int. Conf. on Data and Software Engineering (Yogyakarta)*

[8] Pelekis N, Theodoulidis B, Kopanakis I and Theodoridis Y 2004 Literature Review of Spatio-Temporal Database Models *Knowledge Engineering Review* **19** (3) p 235–274

[9] Güting R H and M. Schneider M 2005 *Moving Object Databases* (San Fransisco: Morgan Kaufmann)