Technical Report RT-122

Linking the Shared Vision Model to the information management strategy

André Plante and Wendy Leger

Mach 2003

For reference:

Plante, A. and W. Leger (2003). Linking the Share Vision Model to the information management strategy. Technical Report MSC Québec – Hydrology RT-122, Environment Canada, Sainte-Foy, prepared for the Plan Formulation and Evaluation Group (PFEG) of the of the International Lake Ontario – St. Lawrence River Study Board (International Joint Commission). 19 pages + Appendix.

RESEARCH TEAM

Environment Canada, Meteorological Service of Canada – Hydrology Section

Coordination, design and report writing	André Plante and Wendy Leger
Programming	Sylvain Martin

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

The Lake Ontario-St. Lawrence River Study Board was established by the International Joint Commission (IJC) to review the regulation of Lake Ontario water levels to address a broader range of interests than were considered in the criteria for regulation included in the International Joint Commissions' 1956 Orders of Approval. The Study Board must consider new criteria and new regulating plans simultaneously and in 2005 present options and recommendations to the IJC for possible amendments to the present Orders of Approval.

The Plan Formulation and Evaluation Group (PFEG) is one of nine Technical Working Group (TWG) established by the Study Board to assist with the study. The job of the PFEG is to aid the Study Board in their decision-making process by providing an integrated and effective system to formulate, evaluate and rank alternative Lake Ontario regulation plans. The PFEG will integrate all work done by the various TWGs to assess impacts of possible changes to the regulation plan on their respective interests. This integration will be accomplished through the use of a multi-objective, multi-stakeholder evaluation procedure which has already been presented to and endorsed by the LOSLRS Study Board. The procedure uses a six-step process whereby composite performance indicators which are functions or metrics related to a particular interest and hydrologic attributes are used. This composes a model known as the Share-Vision Model (SVM). The Shared Vision Model is used to describe the study model being developed to integrate the results from each of the Technical Work Groups (TWGs) in one place so that various regulation plans can be run through an evaluation process and the results can be compared between interests and locations. The primary input into the Shared Vision Model from the other TWGs will be the relationships between a set number of performance indicators identified by each of the TWGs for their interest. A performance indicator is some measure of impact to an interest. For example, the Coastal TWG will use erosion damages in terms of dollars as one of their performance indicators. All the TWGs are working to determine the most scientifically accurate assessment of the relationship between different water levels and flows across the study area and over time to their chosen performance indicators. This relationship, whether it is in terms of a

stage/damage curve or some other mathematical formula, is what will go into the Shared Vision Model for the evaluation process. The TWGs will be generating a substantial amount of data/information/knowledge as they work towards developing these relationships. The PFEG will need to integrate all of this information in order to evaluate and determine if a revised plan performs better than the existing plan or than the other plans proposed.

While the PFEG is responsible for integrating the available information in the evaluation process, the Information Management TWG (formerly the Common Data Needs TWG) was charged with the development and implementation of an Information Management Strategy (IMS). With the assistance of a contractor, Pangaea Information Technologies, the IMS team has conducted a comprehensive Needs Assessment (NA) and hosted two workshops to aid in the formulation of the IMS (Pangaea, 2001). The Pangaea Study concluded that to achieve the goals of the study in terms of Information Management, interoperability standards needed to be respected to ensure integration and connectivity to other systems, and accommodate other technologies such as geospatial web services.

Consequently, this document presents how the PFEG intends to link the SVM with the IMS. A conceptual analysis is performed to determine the best possible way to achieve this goal. The creation of the resulting physical data model will be done. The proposed model follows the pattern of a typical relational database with the implementation of entities (layers) and relational constraints between entities.

2 Shared-Vision Model Integration in the IMS

2.1 Vision

The SVM being developed for the study is a tool to assist the Study Board in their decision-making process. It is not an off-the-shelf model, but rather is built from scratch with the help of each of the TWGs, and it must be easy to use so that all parties can develop and evaluate their own ideas on managing the regulation of Lake Ontario. The vision is for the SVM isfor it to be a layered model which provides a simple interface which permits data mining to allow the user to search out the details and complexities behind the evaluation results.

2.2 Design rules

To meet the SVM/PFEG vision outlined above, the following design rules will be followed to produce the conceptual analysis and resulting physical data models;

- This work must be done in respect of a global architecture described in the Technical Report RT-123 (Plante and Martin, 2003) as well as provide sufficient flexibility to be scalable to permit growth as the study progresses.
- The architecture must permit access to underlying metadata and information linked with any given performance indicator.
- Dynamic visualization capabilities must be incorporated for impacts of performance indicators of resulting water level as a snap shot, but not for the entire time series.
- Maximum use of the internet is necessary to address the large variety of users; general public, researchers and study participants.
- Automated distribution, which includes such things as spatial web services, must be supported to favour the maximum use of the opportunity offered by the internet.

- The work done for designing this data model must be documented to permit easy maintenance and technology transfer.

2.3 Overall Architecture

The overall architecture showing the interaction between the SVM and the remaining components of the IMS can be seen in Figure 1. The PIs are the privileged link between the SVM and the information produced by all of the TWGs.

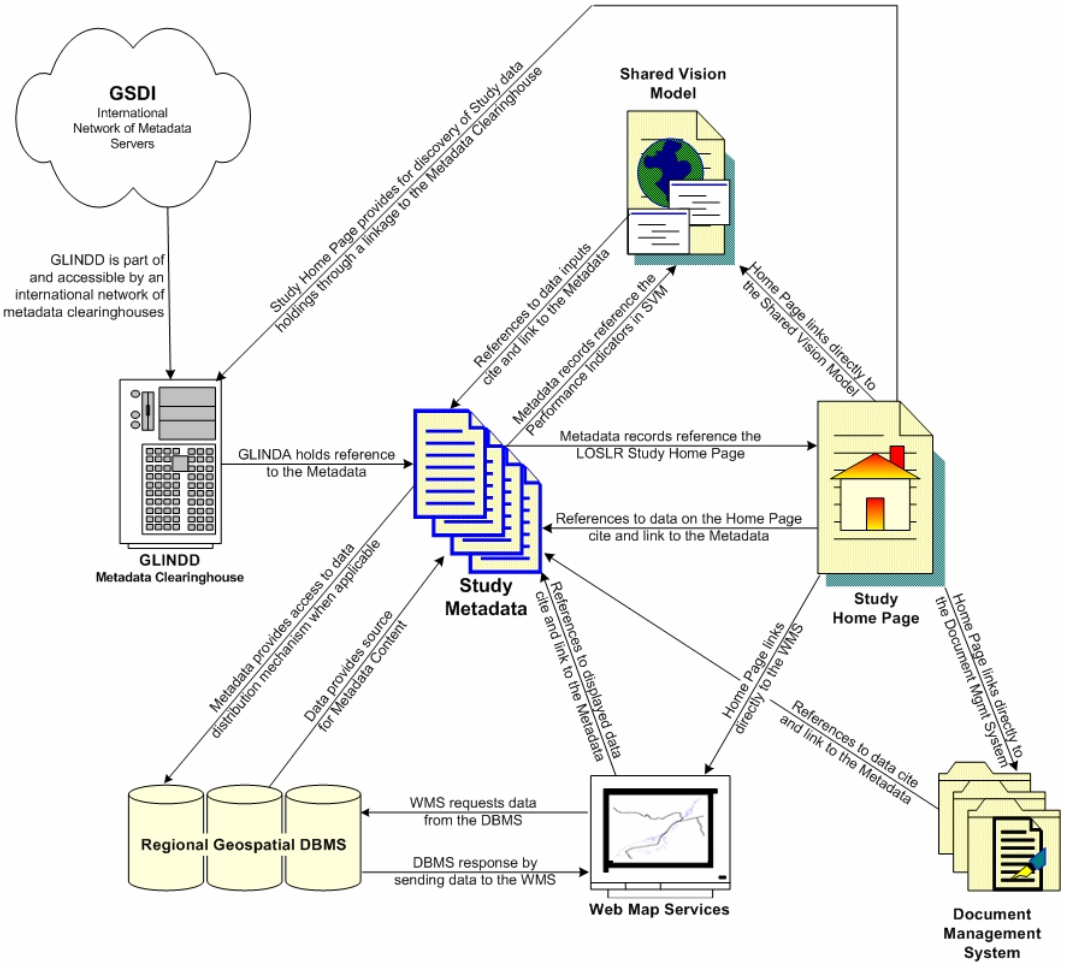

Figure 1: Overall architecture.

3 Conceptual Analysis

Beyond many belief and techniques that follow fashion, conceptual data modeling consists in a rigorous representation of what is stable and hidden behind the apparent variety which constructs our reality. This is captured in a model, which is a representation of reality, using tables, identification attributes and relationships.

Data to be captured in this Conceptual Data Model

Conceptually, links between the SVM, the PIs and the TWGs should be identified and included. This will permit the end users to drill down in the SVM results to gain understanding of the answer gotten. The data that comprises the SVM such as PIs, TWGs and plan results must be included in the model as well. Figure 2 summarizes the addition of needed tables to the model in order to store the desired information. Some of these tables will need to be defined as the SVM matures but we can already see their location in the overall IMS. The data being captured in each table is described in the Table 1.

Table 1: Information layer name and description.

Information layer name	**Information layer description**
TWG_Identification	Task Work Group identification and description
Plan_Identification	Water management plan identification and description
PI_ Identification	TWG developed Performance Indicator identification and description
Plan_Results	Water management plan results to be saved for further use, identification and description
PI _Function	Performance Identification curves to be used by the SVM

Tables

In order to store the information needed by the SVM and meet the vision and objectives stated previously, a total of 5 tables are necessary. The five table are

TWG_IDENTIFICATION, PLAN_IDENTIFICATION, PI_ IDENTIFICATION, PLAN_RESULTS and PI_FUNCTION. The TWG_IDENTIFICATION table should be considered a descriptive table in the spirit of the following section.

Descriptive and identification attributes

The Information Management TWG (IMTWG) has developed the basis for a distributed database. This includes two tables that are common to the all study information layers. These tables are METADATA_CATALOG and TYPE_DEFINITION. The METADATA_CATALOG table hold the metadata information needed to identify each piece of data in the database. This table contains a unique identifier and the location and name of the XML file which contains the metadata produced by the responsible authority for the data. An additional attribute will be added to this table to permit TWG identification. This is necessary to permit drilling of information from the SVM.

The TYPE_DEFINITION table hold a description of the data itself for each entry in the database. The use of descriptor is necessary because the database uses a “per line” approach. This type of approach will permit long term database scalability.

The TWG_IDENTIFICATION table holds the name and description of each of the TWG, in both French and English.

Relational Constraints

In the conceptual model, a relational constraint is shown by an arrow which starts from the identifying attribute in the child table and ends in the mother table at the attribute of the same name. The relational constraints between table TYPE_DEFINITION and each layer table are examples of this relationship. The table containing all data types definition (TYPE_DEFINITION) is the mother table and the data layer table is the child table. The attribute that uniquely identifies data types in the mother table is a primary key in the mother table. The same attribute appears in the data layer table where it plays the role of a foreign key. In order to link this table to the mother table, the layer table must also have

this identification to refer to the table where the data type is explained. The same type of relation exists between each layer table and the table METADATA_CATALOG. Therefore, each of the information layer tables have relations with two other tables, TYPE_DEFINITION and METADATA_CATALOG. The TWG_IDENTIFICATION table is a parent table for table contained in the conceptual model developed to hold information needed by the SVM and the PFEG.

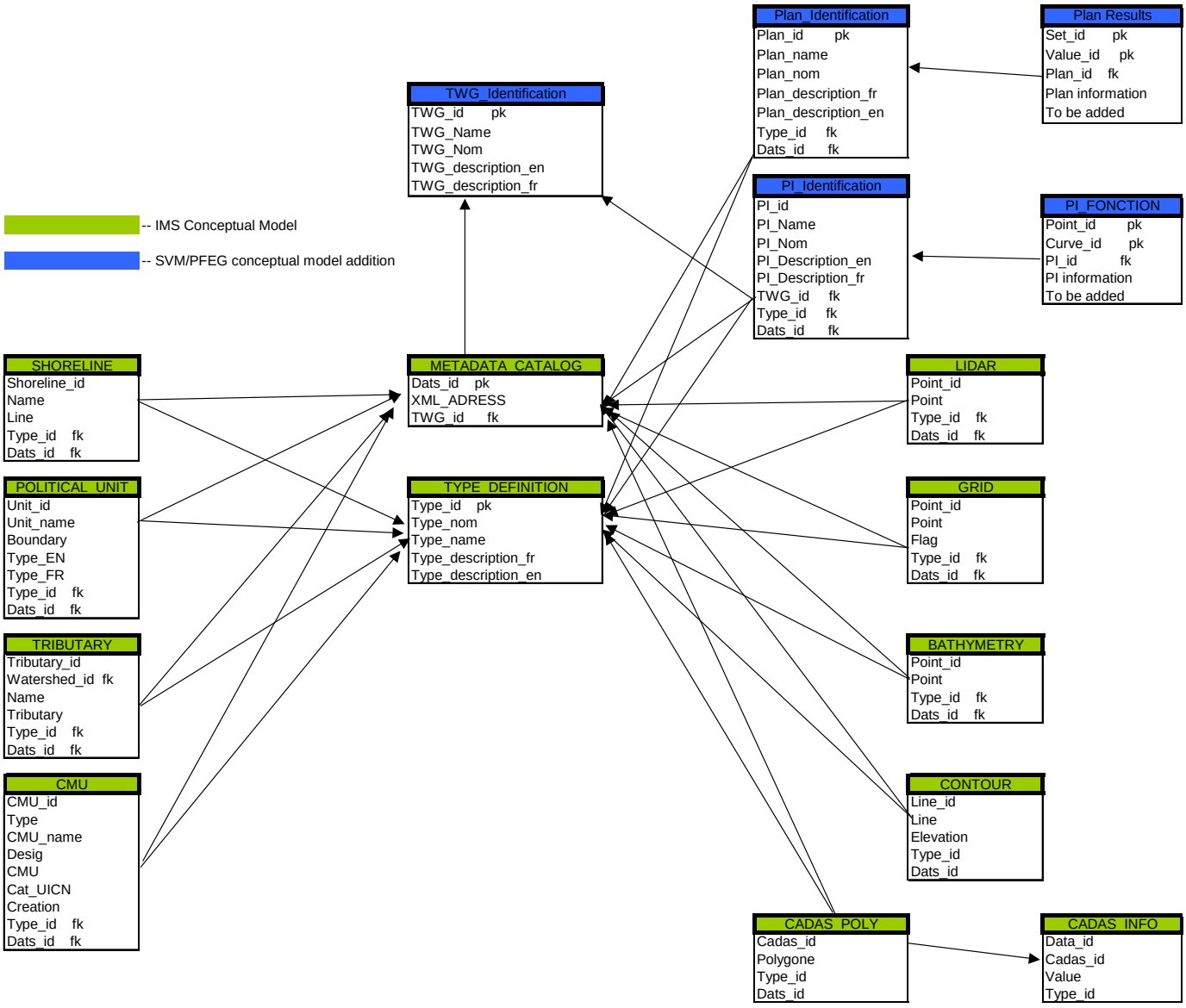

Figure 2: Conceptual data model including SVM/PFEG updates.

4 Resulting Physical Data Model

The physical data model is derived from the conceptual data model. It, in turn, will be used to generate the necessary creation scripts. For each table, attributes are presented with a description of their data type and length. Other characteristics are also shown like primary or foreign keys and the null possibility(absent value). A description of each attribute is done and associated constraints are detailed. The tables are presented in their order of creation within the database. This is necessary to respect relational constraints among tables.

Data types used here follow the Oracle 9i SQL data types implementation. The two following types were used:

Number: Numerical type (real or integer according to the size and precision)
Varchar2: Variable length character chain type.

4.1 Table TWG_IDENTIFICATION

This table contains a description of the different TWG. Some of the tables of the model contain a foreign key on the attribute TWG_id. For this reason, this table is created first.

Atribute	**Type**	**Length**	**Null**	**Indexed**	**Key**
TWG_id	Number	3	No	Yes	Primary
TWG_nom	Varchar2	25	No	No	
TWG_name	Varchar2	25	No	No	
TWG_description_fr	Varchar2	250	No	No	
TWG_description_en	Varchar2	250	No	No	

Attribute description:
TWG_id: Unique identification number for a data TWG.
TWG_name: The name of the data TWG in English.
TWG_nom: The name of the data TWG in French.
TWG_description_fr: Short description of the data TWG in French.
TWG_description_en: Short description of the data TWG in English.

Constraints:

- Each TWG_id must be unique.

4.2 Table METADATA_CATALOG

Each table contains a foreign key on the attribute Dats_id from the table METADATA_CATALOG. Before one inserts a data set table, an entry must appear in the METADATA_CATALOG table that refers to metadata information. This insures that each piece of data in the database is linked with it's metadata. This table was created when the database and base layers where created by the IMTWG. One attribute must however be added to it, TWG_id, to meet the needs of the PFEG.

Atribute	**Type**	**Length**	**Null**	**Indexed**	**Key**
Dats_id	Number	3	No	Yes	Primary
XML_ADRESS	Varchar2	100	No	No	
TWG_id	Number	3	No	Yes	Foreign

Attribute description:
Dats_id: Unique identification number for a data type.
XML_ADRESS : Adress of the XML file in which is described this datatype.
TWG_id : TWG identifier.

Constraints:

- Each Dats_id must be unique.
- The foreign key TWG_id refers to the primary key TWG_id in the TWG_IDENTIFICATION table.

4.3 Table PLAN_IDENTIFICATION

This table can contain water management plan identification and description. The name and identification of the plan are taken into account for both French and English.

Atribute	**Type**	**Length**	**Null**	**Indexed**	**Key**
Plan_id	Number	10	No	Yes	Primary
Plan_name	Varchar2	50	No	No	
Plan_nom	Varchar2	50	Yes	No	
Plan_description_en	Varchar2	256	Yes	No	
Plan_description_fr	Varchar2	256	Yes	No	
Type_id	Number	10	No	No	Foreign
Dats_id	Number	10	No	No	Foreign

Attribute description:

Plan_id: Unique identification number for each water management plan.
Plan_name: Name of water management plan in English.
Plan_nom: Name of water management plan in French.
Plan_description_en: Description of plan in English.
Plan_description_fr: Description of plan in French.
Type_id: Data type identification number.
Dats_id: Metadata identifier.

Constraints:

- Each Plan_id number must be unique.
- The foreign key Type_id refers to the primary key Type_id in the TYPE_DEFINITION table.
- The foreign key Dats_id refers to the primary key Dats_id in the METADATA_CATALOG table.

4.4 Table PI_IDENTIFICATION

This table can contain Performance Indicator (PI) identification and description. The name and identification of the PI are taken into account for both French and English. The TWG responsible for producing this PI is also linked for each of data. This will permit the drilling of information from the SVM.

Atribute	**Type**	**Length**	**Null**	**Indexed**	**Key**
PI_id	Number	10	No	Yes	Primary
PI_name	Varchar2	50	No	No	
PI_nom	Varchar2	50	Yes	No	
PI_description_en	Varchar2	256	Yes	No	
PI_description_fr	Varchar2	256	Yes	No	
Type_id	Number	10	No	No	Foreign
Dats_id	Number	10	No	No	Foreign
TWG_id	Number	3	No	Yes	Foreign

Attribute description:
PI_id: Unique identification number for PI.
PI_name: Name of PI in English.
PI_nom: Name of PI in French.
PI_description_en: Description of PI in English.
PI_description_fr: Description of PI in French.
Type_id: Data type identification number.
Dats_id: Metadata identifier.
TWG_id : TWG identifier.

Constraints:

- Each PI_id number must be unique.
- The foreign key Type_id refers to the primary key Type_id in the TYPE_DEFINITION table.
- The foreign key Dats_id refers to the primary key Dats_id in the METADATA_CATALOG table.
- The foreign key TWG_id refers to the primary key TWG_id in the TWG_IDENTIFICATION table.

4.5 Table PLAN_RESULTS

This table can contain water management plan results identification and description. This table, in it's present form, is more of a placeholder than a real table. It will be fully developed as the SVM comes to maturity in the years to come.

Atribute	**Type**	**Length**	**Null**	**Indexed**	**Key**
SET_id	Number	10	No	Yes	Primary
Value_id	Number	3	Yes	Yes	Foreign
Plan_id	Number	10	Yes	Yes	Foreign
Plan information To be added based on SVM development next fiscal					

Attribute description:
SET_id: Unique identification number for each water management plan results.
Type_id: Type of data identification
Plan_id: Water management plan identification.

Constraints:

- Each SET_id number must be unique.
- The foreign key Type_id refers to the primary key Type_id in the TYPE_DEFINITION table.

4.6 Table PI_FUNCTION

This table can contain Performance Indicator (PI) curves. This table, in it's present form, is more of a placeholder than a real table. It will be fully developed as the SVM comes to maturity in the years to come.

Atribute	Type	Length	Null	Indexed	Key
SET_id	Number	10	No	Yes	Primary
Curve_id	Number	10	Yes	Yes	Foreign
PI_id	Number	10	Yes	Yes	Foreign
Plan information To be added based on SVM development next fiscal					

Attribute description:
SET_id: Unique identification number for each water management plan results.
Type_id: Type of data identification.
PI_id: Identification of Performance Indicator curves.

Constraints:

- Each SET_id number must be unique.
- The foreign key PI_id refers to the primary key PI_id in the PI_IDENTIFICATION table.

4.7 Table TYPE_DEFINITION

This table contains a description of all the different types of data that are held in each of the layers present in the database. It was created by the IMTWG for all participants to use. It will not be recreated by the PFEG, appropriate type definition will be added to this table. Each of the tables of the model contain a foreign key on the attribute Type_id. For this reason, this table must be present for the PFEG scripts to execute correctly. The definition of this table was added to the present report solely in an effort to provide a complete picture to the reader.

Atribute	Type	Length	Null	Indexed	Key
Type_id	Number	3	No	Yes	Primary
Type_nom	Varchar2	25	No	No	
Type_name	Varchar2	25	No	No	
Type_description_fr	Varchar2	250	No	No	

Type_description_en	Varchar2	250	No	No	

Attribute description:
Type_id: Unique identification number for a data type.
Type_name: The name of the data type in English.
Type_nom: The name of the data type in French.
Type_description_fr: Short description of the data type in French.
Type_description_en: Short description of the data type in English.

Constraints:

- Each Type_id must be unique.

5 Database Physical Model Creation Scripts

Creation scripts are necessary to transfer the structure produced through the conceptual and physical data model work in the database. These are written in SQL-DDL using a normal text editor. Table 1 depicts creation script names and their individual roles as well as the order in which they must be executed.

Table 2: Creation scripts name and role in the database creation.

No	Creation script name	Role
1-	SC_TABLESPACE_TWG.SQL	Creates tablespaces for new tables
2-	SC_TA_TWG.SQL	Creates the 5 tables needed
3-	SC_MODIF_META_CAT.SQL	Adds needed attribute to METADATA_CATALOG table

General documentation

RELATIONS

Each of the tables that contain information layers contains a data type description field. These data types are described in the TYPE_DEFINITION table of the database. This creates a “relationship” between each table representing the data layers and the TYPE_DEFINITION table. The SVM will have two type of data, Plans and Performance indicators. These also have a metadata description identification number which is stored in the METADATA_CATALOG table.

In a relational model context, this is implemented with a primary key in the mother table, for example represented here by the attribute Type_id in the TYPE_DEFINITION table; and a foreign key in the child tables, represented also by the attribute Type_id in each of the different layer tables. The same holds true for the attributes Dats_id and TWG_id.

UNIQUE CONSTRAINTS

Some of the tables have a unique constraint on their identification number attribute (ex. PI_id has a unique constraint). This is used to avoid duplication. A sequence generator is created for each of these attributes to help create the values given to identification numbers as unique values. Additionally, each insertion in these tables must use the NEXTVAL pseudocolumn. The next SQL sentence inserts a row in the table SHORELINE. This table uses the Sequence SHORELINE_SEQ.

INSERT INTO TABLE USR_CDN.SHORELINE VALUES (SHORELINE_SEQ.NEXTVAL, 'St. Lawrence river shore', MDSYS.SDO_GEOMETRY (2002, 82196…….

CREATION SCRIPT ORDER

Tables are created by existing users. Therefore users have to be created before the table since tables become part of a user's schema. Next, in order to respect relational constraints, mother tables (table containing information on which other tables depend) have to be created before child tables. The creation order suggested in table 1 respects relational constraints and user creation.

CREATION SCRIPT FEATURES

Each creation script drops every object before recreating them. This way, if one has to run a given script more than once, objects don't have to be dropped individually before running the script again. However, if data is present in the table being dropped, it will be deleted.

At the beginning of each table creation script, the user is granted sufficient rights to create the table. These rights are revoked at the end of each script. This way, one can connect with the user name but cannot create objects.

VIEWS AND TABLES

For each layer table, a view is created and gives access to all table attributes with the exception of the Type_id and Dats_id. These views avoid end-users seeing management

information and also avoid data updates by end users. Users may have different rights on a view or a table. Some users have the right to select, or insert, or delete or update. For example, the data set manager can select, insert, delete or update rows in a table or a view, but an end-user may only have the right to select information on a view.

In this case, end-users will have selection rights on created views only, to avoid any unwanted updates on tables.

6 Conclusion

The work accomplished within this report includes:

- The description of a vision and associated design objectives to meet the needs of the Shared Vision Model (SVM) and of the PFEG
- The production of conceptual and physical data models that meet the needs expressed by the defined vision and objectives.
- The production of the SQL scripts needed the tables and relational constraints needed to organize the different data sets.

As the study progresses and the SVM matures, the following changes to the database are expected:

- The Plan_Results table will be defined to meet the needs of the SVM.
- The PI_FUNCTION table will be defined to store the curves or surface or address the needs of the SVM.
- Modification may occur to the SVM/PFEG tables to permit unexpected information drilling from the SVM
- Relational Constraints will be added to the data model to enforce design rules.

7 References

Pangeae (2002). *Information Management Strategy for the International Joint Commission Lake Ontario-St.Lawrence River Study*, May 2002. Pangeae Information Technologies, Ltd. 130 p.

Plante, A. and Martin, S. (2003). *Relational database development : conceptual and physical models with creation scripts.* Environnement Canada, Service météorologique du Canada, Rapport technique RT-123, Sainte-Foy. 43p

CDNTWG (2002). *Short-Term GIS Guidelines*. Produced by the Common Data Need Task Work Group of the International Joint Commission Lake Ontario-St.Lawrence River Study, March 21, 2002. 7 p.

APPENDIX A – CREATION SCRIPTS

SC_TABLESPACE_TWG.SQL

```
--*****************************************************
--*              CREATION SCRIPT
--*                 FOR TABLESPACE TWG
--*****************************************************

CONNECT SYS/PLANTE003@BD3 AS SYSDBA;

--***********************************************************
--*        TABLESPACE DESTRUCTION
--***********************************************************

DROP TABLESPACE TS_TWG INCLUDING CONTENTS AND DATAFILES;



--***********************************************************
--*        TABLESPACE CREATION
--***********************************************************

CREATE TABLESPACE TS_TWG DATAFILE 'C:\ORADATA\TS_TWG.DAT' SIZE 10 M
          AUTOEXTEND ON MAXSIZE UNLIMITED
          EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K;


PROMPT *******************************
PROMPT *  END OF CREATION SCRIPT  *
PROMPT *******************************
--*****************************************************************************
--*                    EOF.                        *
--*****************************************************************************
```

SC_TA_TWG.SQL

```
--*****************************************************
--*          CREATION SCRIPT FOR
--*          TABLES USR_COMMUN.TWG_IDENTIFICATION
--*          USR_COMMUN.PLAN_IDENTIFICATION
--*          USR_COMMUN.PI_IDENTIFICATION
--*          USR_COMMUN.PLAN_RESULTS
--*          USR_COMMUN.PI
--*****************************************************


--*********  CONNECTION BY THE DBA
CONNECT SYS/PLANTE003@BD3 AS SYSDBA;

--******** GRANTS TO THE TABLE CREATOR AND CONNECTION
GRANT RESOURCE, DBA ,CREATE TABLE TO USR_COMMUN;

CONNECT USR_COMMUN/COMMUN@BD3;

--*********************************************************
--*          VIEW AND TABLE DESTRUCTION
--*********************************************************
DROP VIEW USR_COMMUN.V_TWG;
DROP VIEW USR_COMMUN.V_PLAN_IDEN;
DROP VIEW USR_COMMUN.V_PI_IDEN;
DROP VIEW USR_COMMUN.V_PLAN_RESULT;
DROP VIEW USR_COMMUN.V_PI;
DROP TABLE USR_COMMUN.PI;
DROP TABLE USR_COMMUN.PLAN_RESULTS;
DROP TABLE USR_COMMUN.PLAN_IDENTIFICATION;
DROP TABLE USR_COMMUN.PI_IDENTIFICATION;
DROP TABLE USR_COMMUN.TWG_IDENTIFICATION;

--*********************************************************
--*          SEQUENCE DESTRUCTION
--*********************************************************
DROP SEQUENCE USR_COMMUN.TWG_ID_SEQ;
DROP SEQUENCE USR_COMMUN.PLAN_ID_SEQ;
DROP SEQUENCE USR_COMMUN.PI_IDEN_SEQ;
DROP SEQUENCE USR_COMMUN.PLAN_RESULT_SEQ;
DROP SEQUENCE USR_COMMUN.PI_ID_SEQ;


--*****************************************************
--*          TABLE CREATION
--*****************************************************


CREATE TABLE USR_COMMUN.TWG_IDENTIFICATION(
          TWG_ID                        NUMBER(10) NOT NULL,
          TWG_NOM                       VARCHAR2(25) NOT NULL,
          TWG_NAME                      VARCHAR2(25) NOT NULL,
```

```
	TWG_DESCRIPTION_FR		VARCHAR2(250) NOT NULL,
	TWG_DESCRIPTION_EN		VARCHAR2(250) NOT NULL,
	CONSTRAINT TWG_IDEN_PK PRIMARY KEY (TWG_ID))
	TABLESPACE TS_TWG;

CREATE OR REPLACE VIEW USR_COMMUN.V_TWG AS
  SELECT TWG_ID, TWG_NOM, TWG_NAME,  TWG_DESCRIPTION_FR,
TWG_DESCRIPTION_EN
		FROM USR_COMMUN.TWG_IDENTIFICATION;


CREATE TABLE USR_COMMUN.PLAN_IDENTIFICATION(
	PLAN_ID			NUMBER(10) NOT NULL,
	PLAN_NOM		VARCHAR2(25) NOT NULL,
	PLAN_NAME		VARCHAR2(25) NOT NULL,
	PLAN_DESCRIPTION_FR		VARCHAR2(250) NOT NULL,
	PLAN_DESCRIPTION_EN		VARCHAR2(250) NOT NULL,
	TYPE_ID				NUMBER(10) NOT NULL,
	DATS_ID				NUMBER(10) NOT NULL,
	CONSTRAINT PLAN_IDEN_PK PRIMARY KEY (PLAN_ID),
	CONSTRAINT PLAN_IDEN_TYPE_DEF_FK FOREIGN KEY(TYPE_ID) REFERENCES
USR_COMMUN.TYPE_DEFINITION(TYPE_ID),
	CONSTRAINT PLAN_IDEN_META_CAT_FK FOREIGN KEY(DATS_ID) REFERENCES
USR_COMMUN.METADATA_CATALOG(DATS_ID)
	)
	TABLESPACE TS_TWG;


CREATE OR REPLACE VIEW USR_COMMUN.V_PLAN_IDEN AS
  SELECT PLAN_ID, PLAN_NOM, PLAN_NAME, PLAN_DESCRIPTION_FR,
PLAN_DESCRIPTION_EN
  FROM USR_COMMUN.PLAN_IDENTIFICATION;



CREATE TABLE USR_COMMUN.PI_IDENTIFICATION(
	PI_ID			NUMBER(10) NOT NULL,
	PI_NOM			VARCHAR2(25) NOT NULL,
	PI_NAME			VARCHAR2(25) NOT NULL,
	PI_DESCRIPTION_FR	VARCHAR2(250) NOT NULL,
	PI_DESCRIPTION_EN	VARCHAR2(250) NOT NULL,
	TWG_ID			NUMBER(8) NOT NULL,
	TYPE_ID				NUMBER(10) NOT NULL,
	DATS_ID				NUMBER(10) NOT NULL,
	CONSTRAINT PI_IDEN_PK PRIMARY KEY (PI_ID),
	CONSTRAINT PI_IDEN_TWG_IDEN_FK FOREIGN KEY(TWG_ID) REFERENCES
USR_COMMUN.TWG_IDENTIFICATION(TWG_ID),
	CONSTRAINT PI_IDEN_TYPE_DEF_FK FOREIGN KEY(TYPE_ID) REFERENCES
USR_COMMUN.TYPE_DEFINITION(TYPE_ID),
	CONSTRAINT PI_IDEN_META_CAT_FK FOREIGN KEY(DATS_ID) REFERENCES
USR_COMMUN.METADATA_CATALOG(DATS_ID)
	)
	TABLESPACE TS_TWG;

CREATE OR REPLACE VIEW USR_COMMUN.V_PI_IDEN AS
  SELECT PI_ID, PI_NOM, PI_NAME, PI_DESCRIPTION_FR, PI_DESCRIPTION_EN
```

```
   FROM USR_COMMUN.PI_IDENTIFICATION;


CREATE TABLE USR_COMMUN.PLAN_RESULTS(
        SET_ID                  NUMBER(10) NOT NULL,
        VALUE_ID                        NUMBER(10) NOT NULL,
        PLAN_ID                         NUMBER(10) NOT NULL,
        CONSTRAINT PLAN_RESULT_PK PRIMARY KEY (SET_ID, VALUE_ID),
        CONSTRAINT PLAN_RES_PLAN_ID_FK FOREIGN KEY(PLAN_ID) REFERENCES
USR_COMMUN.PLAN_IDENTIFICATION(PLAN_ID)
        )
        TABLESPACE TS_TWG;

CREATE OR REPLACE VIEW USR_COMMUN.V_PLAN_RESULT AS
   SELECT SET_ID, VALUE_ID, PLAN_ID
   FROM USR_COMMUN.PLAN_RESULTS;



CREATE TABLE USR_COMMUN.PI(
        POINT_ID                        NUMBER(10) NOT NULL,
        CURVE_ID                        NUMBER(10) NOT NULL,
        PI_ID                           NUMBER(10) NOT NULL,
        CONSTRAINT PI_PK PRIMARY KEY ( POINT_ID, CURVE_ID),
        CONSTRAINT PI_PI_IDEN_FK FOREIGN KEY(PI_ID) REFERENCES
USR_COMMUN.PI_IDENTIFICATION (PI_ID)
        )
        TABLESPACE TS_TWG;

CREATE OR REPLACE VIEW USR_COMMUN.V_PI AS
   SELECT POINT_ID, CURVE_ID, PI_ID
   FROM USR_COMMUN.PI;



--**** DROITS DE REFERENCE A USR_CDN
GRANT REFERENCES ON USR_COMMUN.TWG_IDENTIFICATION TO USR_CDN;
GRANT REFERENCES ON USR_COMMUN.PLAN_IDENTIFICATION TO USR_CDN;
GRANT REFERENCES ON USR_COMMUN.PI_IDENTIFICATION TO USR_CDN;
GRANT REFERENCES ON USR_COMMUN.PLAN_RESULTS TO USR_CDN;
GRANT REFERENCES ON USR_COMMUN.PI TO USR_CDN;


-- ***  SEQUENCE CREATION FOR IDENTIFIERS

CREATE SEQUENCE USR_COMMUN.TWG_ID_SEQ
    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
    NOCYCLE
    CACHE 100;

CREATE SEQUENCE USR_COMMUN.PLAN_ID_SEQ
    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
```

```
    NOCYCLE
    CACHE 100;

CREATE SEQUENCE USR_COMMUN.PI_IDEN_SEQ
    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
    NOCYCLE
    CACHE 100;

CREATE SEQUENCE USR_COMMUN.PLAN_RESULT_SEQ
    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
    NOCYCLE
    CACHE 100;

CREATE SEQUENCE USR_COMMUN.PI_ID_SEQ
    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
    NOCYCLE
    CACHE 100;



-- RECONNECTION BY THE DBS
CONNECT SYS/PLANTE003@BD3 AS SYSDBA;

--*************************************************************
--*         RIGHTS REVOCATION FROM THE CREATOR
--*************************************************************

REVOKE RESOURCE, DBA, CREATE TABLE  FROM USR_COMMUN;

-- WE ACCEPT THE TRANSACTIONS
COMMIT;

PROMPT *********************************
PROMPT *  LE SCRIPT SQL EST TERMINÉ  *
PROMPT *********************************

--*****************************************************************************
--*                      FIN DU FICHIER SQL.                          *
--*****************************************************************************
```

SC_MODIF_META_CAT.SQL

```
--*****************************************************
--*              SCRIPT FOR MIDIFICATION IN
--*                  TABLE USR_COMMUN.METADATA_CATALOG
--*****************************************************

CONNECT SYS/PLANTE003@BD3 AS SYSDBA;

ALTER TABLE USR_COMMUN.METADATA_CATALOG ADD TWG_ID NUMBER(10) NOT
NULL;

ALTER TABLE  USR_COMMUN.METADATA_CATALOG ADD CONSTRAINT
META_CAT_PLAN_ID_FK FOREIGN KEY (TWG_ID)
        REFERENCES USR_COMMUN.TWG_IDENTIFICATION (TWG_ID);

COMMIT;

PROMPT ********************************
PROMPT *  END OF CREATION SCRIPT  *
PROMPT ********************************
--*****************************************************************************
--*                    EOF.                       *
--*****************************************************************************
```