

УДК 623.644

В. Е. Лісцин, А. А. Побережний, С. А. Горєлишев

МОДЕЛЮВАННЯ ПРОСТОРОВИХ ДАНИХ ДЛЯ ЕЛЕКТРОННИХ КАРТ МІСЦЕВОСТІ, ЯКІ ВИКОРИСТОВУЮТЬСЯ ДЛЯ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ ВНУТРІШНІХ ВІЙСЬК МВС УКРАЇНИ

Розглянуто деякі особливості створення електронних карт для інформаційного забезпечення внутрішніх військ МВС України. Проаналізовано сучасні підходи, що використовуються під час опису просторових даних і подання метайнформації на основі геоінформаційних технологій.

Постановка проблеми. Незважаючи на стрімке зростання технологій в області геоінформаційних систем (ГІС), універсальні продукти, подібні до ArcGIS, MAPINFO або ERDAS IMAGINE, не завжди повною мірою задовольняють вимоги користувачів і розробників електронних карт. Річ саме в універсальності великих програмних пакетів, прагненні запропонувати стандартні рішення “на всі випадки життя”. Закритість базових технологій у процесі роботи з просторовими об’єктами, принцип “чорного ящика” під час виконання аналізу даних примушують кінцевого користувача або сліпо довіряти запропонованому програмному продукту, або шукати альтернативні джерела з відкритим кодом. У багатьох областях, і особливо у військових додатках, необхідна більш керована оболонка управління електронною картою, що піддається аналізу і водночас гарантовано захищена від навмисних або ненавмисних вторгнень в програмний код.

Крім цього, користувачеві не потрібна вся потужність “великої” програмної оболонки, більшість функцій якої є надмірними під час вирішення конкретних задач. Поділ універсальної оболонки на модулі допомагає у цьому випадку не завжди. Дуже часто, працюючи з одним модулем, користувач має потребу у додатковій функціональності, хоча б невеликій, але реалізованій в іншому модулі. Тому не зменшується інтерес до розробки спеціалізованих програмних оболонок для роботи з електронними картами, що реалізують вузький, обмежений жорсткими інструкціями діапазон задач.

Окрім атрибутивної інформації під час складання електронної карти необхідна інформація про дані самої карти. До таких даних можуть бути віднесені: назви шарів карти, кольори, що використовуються для відображення шарів, назви шрифтів, які застосовуються для підпису об’єктів, мінімальний і максимальний масштаби відображення шарів, елементи електронного класифікатора, назви проєкцій карти та ін. Для даних такого роду є спеціальний термін – *метайнформація* (інформація про інформацію).

Аналіз останніх досліджень і публікацій. У кожного розробника, що працює з просторовими даними, організованими у вигляді файлової структури, рано чи пізно виникала думка про необхідність зовнішньої бази даних, що містить метайнформацію. У розробників ГІС-додатків виникла ідея: навіщо поділяти просторову інформацію і метайнформацію?

Так з’явилися просторові дані, що зберігаються безпосередньо у СУБД. На сьогодні широко застосовуються кілька форматів зберігання даних.

1. Просторові типи даних у СУБД MS-SQL 2008 (перша версія). За допомогою методів, передбачених для просторових типів даних GEOGRAPHY та GEOMETRY в SQL Server 2008, можна писати код на T-SQL, що реалізує операції з просторовими даними, наприклад, виконує пошук точок перетину двох об’єктів карти або визначає відстані між населеними пунктами.

2. Просторові типи даних у СУБД ORACLE.

3. Технологія SDE фірми ESRI з використанням бази геоданих.

Технологія SDE об’єднує набір бібліотек для роботи з просторовими даними геоінформаційної системи ArcGIS. Зберігання самих шарів карти та їх описової інформації організовано в локальних або розподілених геобазах даних. А доступ і безпека інформації забезпечуються засобами серверних СУБД Oracle або MS-SQL. SDE дозволяє проводити контроль даних уже на етапі створення об’єкта. Для цього на рівні реляційних таблиць здійснюється контроль значень, що вводяться, який реалізований у механізмі вбудованих процедур СУБД. Окрім цього, маючи можливість об’єднати різнотипні об’єкти у *набір (feature dataset)*, користувач вирішує проблему єдиної проєкції та єдиної системи координат, а також контролює вихід шару за межі координат усієї карти.

Спеціалізовані оболонки ГІС у кращому разі реалізують обмежений доступ до геобаз даних за допомогою SDE з можливістю тільки читання інформації. А в більшості інших випадків самостійно моделюють за допомогою реляційних таблиць списки шарів карти, параметри шрифтів, підписів і

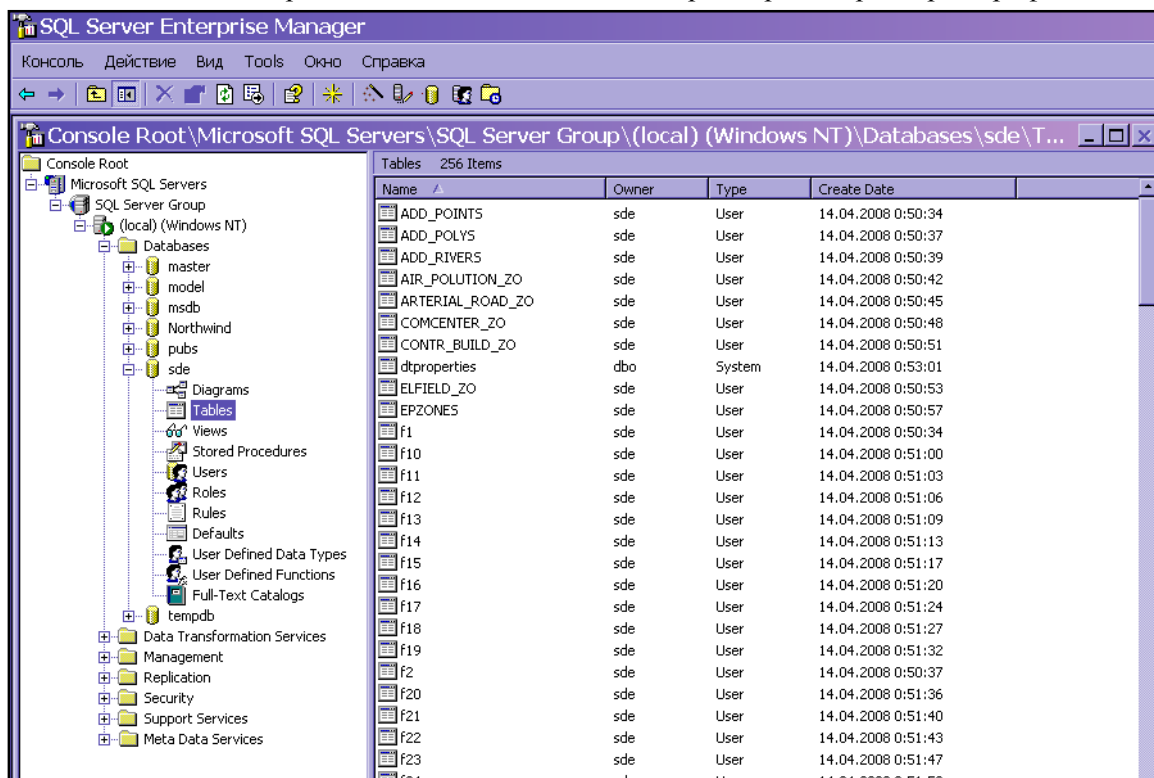


Рис. 1. Об'єкти геобаз даних з ім'ям sde на сервері MS SQL

символів для об'єктів карти, параметри і каталоги зберігання географічних даних, текстові рядки атрибутивних фільтрів, посилання і зв'язані документи, електронні класифікатори та іншу інформацію про електронну карту.

Такі розробки, як правило, потребують залучення до роботи великих колективів програмістів, погодження чималих обсягів програмного коду. Важливим чинником є і швидкість розроблення, адже за довгий час неефективної, повільної роботи багато технологій встигають застаріти. Тому для створення спеціалізованої оболонки, яка працює з електронними картами, потрібен сучасний і потужний засіб розроблення, що підтримує всі новітні технології роботи з даними і може забезпечити функціональність ГІС.

Мета статті – проаналізувати можливість використання сучасних методів моделювання просторових даних електронних карт місцевості, які використовуються для інформаційного забезпечення внутрішніх військ МВС України.

Виклад основного матеріалу. Проаналізуємо деякі підходи, реалізовані у Visual Studio, які дають змогу ефективно використовувати даний продукт у процесі розроблення великих, складних проєктів, пов'язаних з геоінформаційними системами.

Мова розмітки XML має унікальні можливості подання й опису просторових даних і набирає з кожним роком дедалі більшої популярності. Уся комп'ютерна платформа .NET Framework, що лежить в основі програмних продуктів Microsoft Visual Studio, широко застосовує цю мову.

Розглянемо проблему опису системних конфігурацій проєктів, що містять електронні карти. Спочатку всі установки програм в операційній системі WINDOWS зберігалися за допомогою звичайних текстових INI-файлів. Наступним, прогресивнішим кроком стало використання реєстру – ієрархічної структури у пам'яті комп'ютера, де кожній програмі відводилася окрема гілка з набором значень кількох фіксованих типів. Застосування такого підходу за наявності у системі великої кількості встановлених програм призводило до неефективного використання комп'ютерної пам'яті і

перевантаженості системних ресурсів. Тому на цей час практично у всіх програмних продуктах намітилася тенденція зберігати системні установки та параметри у XML-файлах.

```
<?xml version="1.0" ?>
<!-- <!DOCTYPE metadata SYSTEM "http://www.esri.com/metadata/esriprof80.dtd" -->
- <metadata xml:lang="ru">
- <Esri>
  <MetaID>{40F8190C-14C6-40B6-A0A8-BD33A59698E1}</MetaID>
  <CreaDate>20091230</CreaDate>
  <CreaTime>12104200</CreaTime>
  <SyncOnce>FALSE</SyncOnce>
  <SyncDate>20091230</SyncDate>
  <SyncTime>12104200</SyncTime>
  <ModDate>20091230</ModDate>
  <ModTime>12104200</ModTime>
</Esri>
- <idinfo>
  <native Sync="TRUE">Microsoft Windows XP Version 5.1 (Build 2600) Service Pack 2; ESRI
  ArcCatalog 9.2.0.1324</native>
- <descript>
  <langdata Sync="TRUE">ru</langdata>
  <abstract>REQUIRED: A brief narrative summary of the data set.</abstract>
  <purpose>REQUIRED: A summary of the intentions with which the data set was
  developed.</purpose>
</descript>
- <citation>
- <citeinfo>
  <origin>REQUIRED: The name of an organization or individual that developed the data
  set.</origin>
  <pubdate>REQUIRED: The date when the data set is published or otherwise made
  available for release.</pubdate>
  <title Sync="TRUE">4_rgn</title>
```

Рис. 2. Фрагмент XML-файлу, що формує набір метаданих для шару полігональних об'єктів, відкритого в ArcCatalog

Не залишилися осторонь і ГІС. Як приклад наведемо один з варіантів опису просторової метайнформації шарів електронної карти, що використовується компанією ESRI у модулі ArcCatalog. Метадані шару в цьому поданні містять географічні координати точок, що визначають просторову протяжність шару, опис проекції електронної карти та іншу інформацію.

Таке подання електронної карти за допомогою XML-файлів характеризується ієрархічною впорядкованістю та наочністю, що дає змогу організувати ефективний міжмашинний обмін даними у мережах Internet.

У чому ж виявляється ефективність такого обміну? Адже формально файли у форматі XML є текстовими файлами, а аналіз рядків тексту завжди за швидкістю програвав спеціалізованим алгоритмам, що працюють з бінарними даними. Ефективність обробки пов'язана з наявністю вбудованих в .NET Framework синтаксичних аналізаторів XML-файлів. Річ у тому, що модуль System.XML платформи .NET Framework містить низку класів, що дозволяють безпосередньо працювати з XML-документами. Одні з них дають змогу отримувати елементи, здійснюючи тільки просту навігацію від початку до кінця XML-документа. Наприклад, абстрактний клас XMLReader є загальним класом, що реалізує однонаправлене читання в режимі "тільки вперед". На практиці використовується нащадок цього класу XMLTextReader, що має кілька конструкторів. Це дає можливість одержувати інформацію з різних джерел – текстового файлу, пам'яті комп'ютера або URL-ресурсу. Остання можливість дозволяє здійснювати обмін XML-документами, що описують просторові дані, безпосередньо через Internet.

Інші класи, використовуючи кеш-пам'ять, дозволяють завантажити відразу весь документ і здійснити довільний доступ до всіх його елементів. Зокрема, таку модель реалізує клас XmlDocument, розбиваючи повністю завантажений документ на елементи, атрибути, коментарі, інструкції обробки та інші складники. Потім створюється XML-документ у вигляді деревоподібної структури, яка складається з окремих вузлів. Кожен вузол сформовано у вигляді відповідної реалізації абстрактного класу XmlNode.

У XML-форматі можна зберігати деяку частку того обсягу інформації, що зазвичай записують у бази даних. При цьому, працюючи на локальному комп'ютері, користувачеві, як мінімум, можна не

думати про створення і підтримку з'єднання із системою управління базами даних. Окрім цього, XML-документ формує наочну структуру даних і за рахунок використання схем забезпечує сувору перевірку типів. Нарешті, завдяки вбудованій підтримці даного формату на рівні Internet-браузерів стають простими технології публікації просторових даних у глобальних та корпоративних мережах.

З іншого боку, реляційна база даних використовує механізми швидкого пошуку й індексації, механізми транзакцій і блокувань, що дозволяє, наприклад, у призначеному для багатьох користувачів середовищі виконати оновлення вибраної групи записів, не змінюючи решти записів таблиці. А після зміни *будь-якого* елемента або атрибута в XML-документі виконується перезапис *усього* файлу.

Усе це вимагає ретельного аналізу під час прийняття рішення про перенесення географічної інформації про об'єкти електронної карти з реляційних таблиць в ієрархічно структуровані XML-документи. Механізми такого перенесення забезпечуються бібліотекою ADO.NET на рівні об'єкта Dataset. Типову структуру бази даних у форматі СУБД ACCESS, яка моделює просторові дані спеціалізованої оболонки, що розробляється в Академії внутрішніх військ МВС України, подано на рис. 3.

VL_BaseLrNames	Імена, псевдоніми і інформаційні коди базових шарів
VL_Classifier_1	Перший класифікатор
VL_Classifier_2	Другий класифікатор
VL_Connections	Список з'єднань, що визначають розташування шарів
VL_Fonts	Шрифти, зареєстровані в програмі
VL_GnrScheme	Схеми генералізації
VL_HotLinks	Список "гарячих" зв'язків, поданих на карті
VL_LabelStyles	Набори стилів для позначок, що використовуються під час підпису шарів
VL_Layers	Список шарів на картах
VL_Maps	Список карт у проекті
VL_RouterLines	Список лінійних сегментів маршрутизатора
VL_RouterNets	Список мереж маршрутизатора
VL_RouterNodes	Список вузлів маршрутизатора
VL_SampsBody	Вміст шаблонів для створення структури нового шейп-файлу
VL_SampsList	Список шаблонів для створення структури нового шейп-файлу
VL_SymbolStyles	Набори стилів для символів, що використовуються під час відображення шарів
VL_System	Системні константи проекту
VL_Users	Список користувачів

Рис. 3. Метаінформація для опису електронної карти (дані подано у вигляді реляційних таблиць СУБД ACCESS)

Розглянемо детальніше одну з реляційних таблиць – VL_Maps, що містить опис карт, створених у межах проекту. Нижче наведено структуру таблиці, що створена за допомогою конструктора СУБД ACCESS.

VL_Maps : таблиця			
	Имя поля	Тип данных	
🔍	MAPIDX	Счетчик	Индекс карты
	MAPNAME	Текстовый	Имяк карты
	MAPINFO	Текстовый	Текстовое описание карты
	MAPLEFT	Числовой	Левая координата текущего экстенда карты (в единицах проекции)
	MAPTOP	Числовой	Верхняя координата текущего экстенда карты (в единицах проекции)
	MAPRIGHT	Числовой	Правая координата текущего экстенда карты (в единицах проекции)
	MAPBOTTOM	Числовой	Нижняя координата текущего экстенда карты (в единицах проекции)
	MAPISPROJ	Числовой	Проекция карты. 1- Гаусс-Крюгер; 0 - географическая проекция WGS84
▶	MAPGKSYST	Числовой	Система и зона для проекции Гаусса-Крюгера.
	MAPGEOSYST	Числовой	Географическая координатная система в основе карты.
	MAPMINSCALE	Числовой	Минимально допустимый масштаб карты (предел увеличения)
	MAPMAXSCALE	Числовой	Максимально допустимый масштаб карты (предел уменьшения)
	MAPBASESCALE	Числовой	Базовый масштаб карты (т.е. масштаб, с которым она создается)
	MAPGNRGROUP	Числовой	Индекс генерализатора (номер группы в таблице VL_GnrScheme)

Рис. 4. Структура таблиці VL_Maps

Простим способом перенесення вмісту цієї таблиці у XML-документ є програмний виклик методу WriteXml об'єкта DataSet [1]. Нижче надається XML-документ, одержаний після такого виклику. Цей документ має два записи, один з яких включає дані по карті Кримського регіону, другий – по карті всієї України масштабу 1 : 200 000. Дані містять назву й опис карти, базовий, мінімальний і максимальний масштаби, поточні координати аркуша карти, що відображається на екрані, та інформацію про картографічну проекцію.

```
<?xml version="1.0" standalone="yes" ?>
<_Karta3_Dts xmlns="http://tempuri.org/_Karta3_Dts.xsd">
  <VL_Maps>
    <MAPIDX>5</MAPIDX>
    <MAPNAME>Крым. Вектр+растр</MAPNAME>
    <MAPINFO>Крым. Векторные слои и несколько растров</MAPINFO>
    <MAPLEFT>596158.79091322</MAPLEFT>
    <MAPTOP>5071210.82747092</MAPTOP>
    <MAPRIGHT>622294.719973482</MAPRIGHT>
    <MAPBOTTOM>5055800.61268711</MAPBOTTOM>
    <MAPISPROJ>1</MAPISPROJ>
    <MAPGKSYST>28466</MAPGKSYST>
    <MAPGEOSYST>4326</MAPGEOSYST>
  </VL_Maps>
  <VL_Maps>
    <MAPIDX>7</MAPIDX>
    <MAPNAME>Украина</MAPNAME>
    <MAPINFO>Вся Украина</MAPINFO>
    <MAPLEFT>28.3244958894362</MAPLEFT>
    <MAPTOP>50.0567688404826</MAPTOP>
    <MAPRIGHT>36.1609786298688</MAPRIGHT>
    <MAPBOTTOM>45.4362373911776</MAPBOTTOM>
    <MAPISPROJ>0</MAPISPROJ>
    <MAPGKSYST>28466</MAPGKSYST>
    <MAPGEOSYST>4326</MAPGEOSYST>
  </VL_Maps>
</_Karta3_Dts>
```

Характерною особливістю наведеного прикладу є те, що кожному запису початкової таблиці відповідає елемент верхнього рівня VL_Maps, який містить інформацію по одній із карт. Елементи другого рівня надають вміст цього запису. Видно, що структура виходить простою і наочною. Ще однією особливістю отриманого результату є те що, структура сформована таким чином, що весь опис в ній виконується тільки за допомогою *елементів XML*. *Атрибути у цьому XML-документі не використовуються* [2].

Проте така однозначність досягається не завжди. Автори не заглиблюватимуться у тонкощі формування кінцевого результату залежно від інструментарію, що використовується. Наведемо тільки ще два приклади. У цих прикладах початкова таблиця VL_Maps розміщена у базі на MS-SQL Server 2005. Характерною особливістю цієї серверної СУБД є розширений синтаксис мови Transact-SQL. Зокрема, доповнюючи текст стандартного SQL-запиту різними варіантами виразу *FOR XML*, можна одержати результат SQL-запиту у вигляді XML-документа [3].

У другому прикладі таблиця у форматі XML будується за допомогою виразу *FOR XML RAW*.

```
SELECT [MAPIDX]
      ,[MAPNAME]
      ,[MAPINFO]
      ,[MAPLEFT]
      ,[MAPTOP]
      ,[MAPRIGHT]
      ,[MAPBOTTOM]
      ,[MAPISPROJ]
      ,[MAPGKSYST]
      ,[MAPGEOSYST]
FROM [Karta3].[dbo].[VL_Maps]
FOR XML RAW
```

Такий формат створює вихідний документ, що складається з набору елементів одного рівня і має загальне ім'я *row*. Кожен такий елемент відповідає одному запису таблиці *VL_Maps* у базі даних. Для кожного стовпця цього запису усередині елемента створюється відповідний атрибут XML-документа.

```
<row MAPIDX="5"
  MAPNAME="Крым. Вектр+растр"
  MAPINFO="Крым. Векторные слои и несколько растров"
  MAPLEFT="5.961587909132200e+005"
  MAPTOP="5.071210827470920e+006"
  MAPRIGHT="6.222947199734820e+005"
  MAPBOTTOM="5.055800612687110e+006"
  MAPISPROJ="1"
  MAPGKSYST="28466"
  MAPGEOSYST="4326"
/>
<row MAPIDX="7"
  MAPNAME="Украина"
  MAPINFO="Вся Украина"
  MAPLEFT="2.832449588943620e+001"
  MAPTOP="5.0056768884048260e+001"
  MAPRIGHT="3.616097862986880e+001"
  MAPBOTTOM="4.543623739117760e+001"
  MAPISPROJ="0"
  MAPGKSYST="28466"
  MAPGEOSYST="4326"
/>
```

У третьому прикладі для надання реляційної таблиці у форматі XML використовується вираз *FOR XML AUTO*.

```
SELECT [MAPIDX]
      ,[MAPNAME]
      ,[MAPINFO]
      ,[MAPLEFT]
      ,[MAPTOP]
      ,[MAPRIGHT]
      ,[MAPBOTTOM]
```

```
,[MAPISPROJ]  
,[MAPGKSYST]  
,[MAPGEOSYST]  
FROM [Karta3].[dbo].[VL_Maps]  
FOR XML AUTO
```

Такий вираз є більш ефективним у тому випадку, якщо необхідно виконати групування результату SQL-запиту по якому-небудь полю [3]. Проте більш важливий сам вид вихідного документа. Він відрізняється від документа у другому прикладі.

```
<Karta3.dbo.VL_Maps MAPIDX="5"  
  MAPNAME="Крым. Вектр+растр"  
  MAPINFO="Крым. Векторные слои и несколько растров"  
  MAPLEFT="5.961587909132200e+005"  
  MAPTOP="5.071210827470920e+006"  
  MAPRIGHT="6.222947199734820e+005"  
  MAPBOTTOM="5.055800612687110e+006"  
  MAPISPROJ="1"  
  MAPGKSYST="28466"  
  MAPGEOSYST="4326"
```

/>

```
<Karta3.dbo.VL_Maps MAPIDX="7"  
  MAPNAME="Украина"  
  MAPINFO="Вся Украина - 1 й вариант"  
  MAPLEFT="2.832449588943620e+001"  
  MAPTOP="5.005676884048260e+001"  
  MAPRIGHT="3.616097862986880e+001"  
  MAPBOTTOM="4.543623739117760e+001"  
  MAPISPROJ="0"  
  MAPGKSYST="28466"  
  MAPGEOSYST="4326"
```

/>

Отже, на трьох прикладах продемонстровано, що навіть в рамках єдиної технології .NET обробляючи одні й ті ж самі дані, можна одержувати такі результати опису просторових метаданих, що істотно різняться, а це означає, що й інтерпретувати таку інформацію можливо по-різному.

Опис же і публікація самих географічних даних у вигляді XML-документа ставлять ще жорсткіші вимоги до вихідного формату. Необхідно забезпечити ефективне подання векторних географічних даних і контроль їх топології, можливість редагування у XML-файлі окремих об'єктів і вузлів без переписування всього вмісту та опис великих масивів растрових зображень.

Отже, можливо дійти висновку щодо необхідності строгої типізації і стандартизації подання метаданих для просторових даних за допомогою мови розмітки, а також розширення стандартної функціональності XML.

Велику роботу у цьому напрямі проводить міжнародна некомерційна організація OGS (Open Geospatial Consortium), яка об'єднує на сьогодні понад 200 організацій, що займаються дослідженнями і розробками в області інформаційних технологій і геоінформаційних систем. Серед них – компанії ESRI, MapInfo, Google. Організацією OGS була розроблена низка стандартів подання просторової інформації і запропонована мова GML (Geography Markup Language), в основі якої лежить розширений і доповнений формат XML-документа. За аналогією до XML-подання

географічна інформація, яка подається за допомогою GML, може бути доповнена схемою, що описує документ.

У форматі GML дані формуються за допомогою спеціальних тегів (tags), що розширює синтаксис XML на географічну інформацію, наприклад, координати або особливості рельєфу. Теги в GML мають формат, узагальнено визначений як `<gml:XXX>`, де елемент "gml" вказує синтаксичному аналізатору, що використовується розширений синтаксис. Таким чином, у XML-документ не вводиться нова сутність, а тільки відбувається розширення існуючих елементів. Це дозволяє використовувати об'єктну модель документа (DOM) під час аналізу GML-файлів [4]. На прикладі покажемо, як за допомогою мови GML моделюється об'єкт, що задає на растровому зображенні замкнуту полігональну область, яка визначається піксельними координатами.

```
<GML:POLYGON>
  <GML:OUTERBOUNDARYIS>
    <GML:LINEARRING>
      <GML:COORDINATES>100,100 120,200 180,150 100,100</GML:COORDINATES>
    </GML:LINEARRING>
  </GML:OUTERBOUNDARYIS>
</GML:POLYGON>
```

Вигляд та подання об'єктів на екрані (колір, товщина і форма лінії та ін.) указуються за допомогою таблиці стилів. Для того щоб за даними GML накреслити карту, необхідно перетворити GML-подання в один із форматів векторної графіки, наприклад VRML або SVG. Це означає, що потрібно пов'язати з кожним друкованим об'єктом стиль графіки (символи, кольори, шрифт, текстуру). Таким чином, як у XML, так і у GML зміст інформації відокремлено від способу її подання.

У GML міститься велика кількість примітивів, що використовуються для побудови просторових моделей. Такі примітиви включають:

- просторовий об'єкт (feature);
- геометричний об'єкт (geometry);
- систему координат (coordinate reference system, CRS);
- топологію;
- час;
- покриття та географічні растри;
- одиниці вимірювань;
- напрями;
- спостереження і вимірювання;
- правила і стилі відображення карт.

У GML визначено, що просторовий об'єкт відрізняється від геометричного об'єкта. Просторовий об'єкт надає фізичну суть – будівлю, ріку, земельну ділянку. У просторового об'єкта може і не бути геометричного змісту. А геометричний об'єкт визначає координатами або регіоном розміщення, і в цьому його основна відмінність від просторового об'єкта. Координати, що описують геометричний об'єкт, можуть бути задані одним із елементів:

- `<GML: COORDINATES>`;
- `<GML: POS>`.

У термінах GML координати об'єкта – це послідовність N чисел, що позначають позицію точки в N-вимірному просторі. Тоді з використанням елемента `<gml: coordinates>` координати деякої точки описуються так:


```
<GML: POINT GML:ID = "IDX5" SRSNAME = "XXX">  
  <GML:COORDINATES> 45.71 34.78</GML:COORDINATES>  
</GML: POINT>
```

У такому поданні механізм об'єктної моделі документа XML формує елемент `coordinates` як єдиний рядок, не виділяючи окремо широту і довготу. Щоб обійти таке обмеження, можна використовувати інший елемент `<gml:pos>`, що визначає положення об'єкта:

```
<GML: POINT GML:ID = "IDX5" SRSNAME = "XXX">  
  <GML:POS DIMENSION = "2"> 45.71 34.78</GML:POS>  
</GML: POINT>
```

Атрибут `srsName` у цьому прикладі задає параметри координатної системи об'єкта. GML не встановлює координатну систему за замовчуванням, якщо така не задана для об'єкта, тому координати у GML завжди задаються у деякій координатній системі. Координатна система визначена в абстрактній специфікації GML і може бути одного із стандартних типів: геоцентрична, географічна, спроектована, растрова (піксельна). Набір даних (`datum`) визначає еліпсоїд, початок відліку та орієнтування координатної системи, пов'язуючи її із Землею.

Модель проєкції карти у GML описується як координатне перетворення з географічної координатної системи, що визначається географічною широтою і довготою, у планарну систему карти. Двовимірна координатна система належить до типу спроектованих систем.

Приклади, наведені вище, демонструють, що мова GML має достатню функціональність для використання її як інструменту опису просторових даних в задачах інформаційного забезпечення внутрішніх військ МВС України.

Однак просто запропонувати нову описову мову без засобів розроблення на його основі було б неправильно. Тому компанія OGC розробила низку стандартних рішень, що використовують стандартну систему обміну повідомленнями на базі XML. Такі рішення, об'єднані у групу веб-служб (`web-services`), стали програмною основою для публікації й обміну просторовими даними в останніх версіях геоінформаційних систем ArcGIS і ERDAS IMAGINE. Наведемо кілька прикладів таких служб.

1. WMS (Web Map Service) – специфікація інтерфейсу картографічних веб-служб. Дозволяє на базі найпростішого протоколу обміну (всього 3 види запитів) надати клієнту доступ через Internet до растрових карт, знімків та зображень.

2. WFS (Web Feature Service) – специфікація інтерфейсу картографічних веб-служб. Дає можливість на базі протоколу обміну надати клієнту доступ до векторних об'єктів, які моделюються на мові GML, і сформувати векторні багаточастинні карти.

3. CAT (Catalog Interface) – специфікація схеми каталогу геоінформаційних ресурсів і протоколів доступу до нього. Дає змогу здійснювати доступ до просторових даних, що зберігаються у файловому вигляді.

4. SF SQL, CORBA, OLE/COM (Simple Features) – задають правила мережного доступу до баз просторових даних.

Висновки

На цей час мова GML є прогресивною технологією опису та подання метаданих. Вона дає суттєві переваги у процесі розроблення просторових даних порівняно із традиційними засобами, такими, як реляційні таблиці СУБД, технології SDE та мова XML. Усі переваги такого підходу реалізовані у рамках єдиної технології .NET.

Отже, розробникам електронних карт для внутрішніх військ МВС України сьогодні доступні сучасні і високоефективні засоби моделювання структур просторових даних, а також публікації й обміну векторною і растровою інформацією.

Використання цих засобів дасть змогу скоротити час програмної розробки елементів геоінформаційної системи внутрішніх військ МВС України, а також ефективно створювати, редагувати і нарощувати нові структури просторових даних та опису інформації по шарах електронної карти.

Список використаних джерел

1. Род Стивенс. Программирование баз данных / Р. Стивенс. – М. : Бином, 2007. – 384 с.
2. Салих Малик. Microsoft ADO.NET для профессионалов / С. Малик. – М. : Вильямс, 2006. – 560 с.
3. Пол Нильсен. SQL Server 2005. Библия пользователя / П. Нильсен. – К. : Диалектика, 2008. – 1232 с.
4. XML для разработчиков-профессионалов.NET / Динар Дальви, Джо Грэй, Бипин Джошн и др. – М. : Лори, 2003. – 642 с.

Стаття надійшла до редакції 17.05.2010 р.